Symbolic Logic

An Accessible Introduction to Serious Mathematical Logic

Tony Roy

version 7.1

September 20, 2015

Preface

There is, I think, a gap between what many students learn in their first course in formal logic, and what they are expected to know for their second. Thus courses in mathematical logic with metalogical components often cast the barest glance at mathematical induction, and even the very idea of reasoning from definitions. But a first course also may leave these untreated, and fail as well explicitly to lay down the definitions upon which the second course is based. The aim of this text is to integrate material from these courses and, in particular, to make serious mathematical logic accessible to students I teach. The first parts introduce classical symbolic logic as appropriate for beginning students; the material builds to Gödel's adequacy and incompleteness results in the last parts. A distinctive feature of the last part is a complete development of Gödel's second incompleteness theorem.

Accessibility, in this case, includes components which serve to locate this text among others: First, assumptions about background knowledge are minimal. I do not assume particular content about computer science, or about mathematics much beyond high school algebra. Officially, everything is introduced from the ground up. No doubt, the material requires a certain sophistication — which one might acquire from other courses in critical reasoning, mathematics or computer science. But the requirement does not extend to particular contents from any of these areas.

Second, I aim to build skills, and to keep conceptual distance for different applications of 'so' relatively short. Authors of books that are entirely correct and precise, may assume skills and require readers to recognize connections and arguments that are not fully explicit. Perhaps this accounts for some of the reputed difficulty of the material. In contrast, I strive to make arguments almost mechanical and mundane (some would say "pedantic"). In many cases, I attempt this by introducing relatively concrete methods for reasoning. The methods are, no doubt, tedious or unnecessary for the experienced logician. However, I have found that they are valued by students, insofar as students are presented with an occasion for success. These methods are not meant to wash over or substitute for understanding details, but rather to expose and

clarify them. Clarity, beauty and power come, I think, by getting at details, rather than burying or ignoring them.

Third, the discussion is ruthlessly directed at core results. Results may be rendered inaccessible to students, who have many constraints on their time and schedules, simply because the results would come up in, say, a second course rather than a first. My idea is to exclude side topics and problems, and to go directly after (what I see as) the core. One manifestation is the way definitions and results from earlier sections feed into ones that follow. Thus simple integration is a benefit. Another is the way predicate logic with identity is introduced as a whole in Part I. Though it is possible to isolate sentential logic from the first parts of chapter 2 through chapter 7, and so to use the text for separate treatments of sentential and predicate logic, the guiding idea is to avoid repetition that would be associated with independent treatments for sentential logic, or perhaps monadic predicate logic, the full predicate logic, and predicate logic with identity.

Also (though it may suggest I am not so ruthless about extraneous material as I would like to think), I try to offer some perspective about what is accomplished along the way. In addition, this text may be of particular interest to those who have, or desire, an exposure to natural deduction in formal logic. In this case, accessibility arises from the nature of the system, and association with what has come before. In the first part, I introduce both axiomatic and natural derivation systems; and in Part III, show how they are related.

Answers to selected exercises indicated by star are provided in the back of the book. Answers function as additional examples, complete demonstrations, and supply a check to see that work is on the right track. It is essential to success that you work a significant body of exercises successfully and independently. So do not neglect exercises!

There are different ways to organize a course around this text. For students who are likely to complete the whole, the ideal is to proceed sequentially through the text from beginning to end (but postponing chapter 3 until after chapter 6). Taken as wholes, Part II depends on Part I; parts III and IV on parts I and II. Part IV is mostly independent of Part III. I am currently working within a sequence that isolates sentential logic from quantificational logic, treating them in separate quarters, together covering all of chapters 1 - 7 (except 3). A third course picks up leftover chapters from the first two parts (3 and 8) with Part III; and a fourth the leftover chapters from the first parts with Part IV. Perhaps not the most efficient arrangement, but the best I have been able to do with shifting student populations. Other organizations are possible!

A remark about chapter 7 especially for the instructor: By a formal system for

reasoning with semantic definitions, chapter 7 aims to leverage derivation skills from earlier chapters to informal reasoning with definitions. I have had a difficult time convincing instructors to try this material — and even been told flatly that these skills "cannot be taught." In my experience, this is false (and when I have been able to convince others to try the chapter, they have quickly seen its value). Perhaps the difficulty is that it is "weird" — none of us had (or needed) anything like this when we learned logic. Of course, if one is presented with students whose mathematical sophistication is sufficient for advanced work, the material is not necessary. But if, as is often the case especially for students in philosophy, one obtains one's mathematical sophistication *from* courses in logic, this chapter is an important part of the bridge from earlier material to later. Additionally, the chapter is an important "take-away" even for students who will not continue to later material. The chapter closes an open question — how it is possible to demonstrate quantificational validity — from chapter 4. But further, the ability to reason closely with definitions is a skill from which students in (sentential or) predicate logic, even though they never go on to formalize another sentence or do another derivation, will benefit both for philosophy and more generally.

Naturally, results in this book are not innovative. If there is anything original, it is in presentation. Even here, I am greatly indebted to others, especially perhaps Bergmann, Moor and Nelson, *The Logic Book*, Mendelson, *Introduction to Mathematical Logic*, and Smith, *An Introduction to Gödel's Theorems*. I thank my first logic teacher, G.J. Mattey, who communicated to me his love for the material. And I thank especially my colleagues John Mumma and Darcy Otto for many helpful comments. In addition I have received helpful feedback from Hannah Roy and Steve Johnson, along with students in different logic classes at CSUSB. I welcome comments, and expect that your sufferings will make it better still.

This text evolved over a number of years starting modestly from notes originally provided as a supplement to other texts. It is now long (!) and perhaps best conceived in separate volumes for parts I and II and for parts III and IV. With the addition of Part IV it is complete for the first time in this version. (But chapter 11, which I never get to in teaching, remains a stub that could be developed in different directions.) Most of the text is reasonably stable, though I shall be surprised if I have not introduced errors in the last part both substantive and otherwise. I apologize for these in advance, and anticipate that you will let me hear about them in short order!

I think this is fascinating material, and consider it great reward when students respond "cool!" as they sometimes do. I hope you will have that response more than once along the way.

T.R.
Fall 2015

Contents

Named Definitions

chapter 13

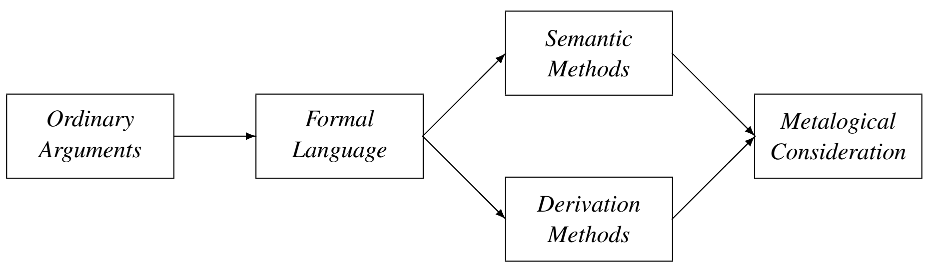
Quick Reference Guides

Part I

The Elements: Four Notions of Validity

Introductory

Symbolic logic is a tool for argument evaluation. In this part of the text we introduce the basic elements of that tool. Those parts are represented in the following diagram.



The starting point is ordinary arguments. Such arguments come in various forms and contexts — from politics and ordinary living, to mathematics and philosophy. Here is a classic, simple case.

(A)
All men are mortal.
Socrates is a man.
____
Socrates is mortal.

This argument has *premises* listed above a line, with a *conclusion* listed below. Here is another case which may seem less simple.

(B) If the maid did it, then it was done with a revolver only if it was done in the parlor. But if the butler is innocent, then the maid did it unless it was done in the parlor. The maid did it only if it was done with a revolver, while the butler is guilty if it did happen in the parlor. So the butler is guilty.

(It is fun to think about this; from the given evidence, it follows that the butler did it!) At any rate, we begin in chapter 1 with an account of success for ordinary arguments

(the leftmost box). This introduces us to the fundamental notions of *logical validity* and *logical soundness*.

But just as it is one thing to know what a cookie is, and another to know whether there is one in the jar, so it is one thing to know what logical validity and soundness are, and another to know whether arguments have them. In some cases, it may be obvious. But others are not so clear. Consider, say, the butler case (B) above, along with complex or controversial arguments in philosophy or mathematics. Thus symbolic logic is introduced as a sort of machine or tool to identify validity and soundness. This machine begins with certain formal representations of ordinary reasonings. We introduce these representations in chapter 2 and translate from ordinary arguments to the formal representations in chapter 5 (the box second from the left). Once arguments have this formal representation, there are different modes of operation upon them. A semantic notion of validity is developed in chapter 4 and chapter 7 (the upper box). And a pair of derivation systems, with corresponding notions of validity, are introduced in chapter 3 and chapter 6 (the lower box). Evaluation of the butler case is entirely routine given the methods of just the first parts from, say, chapter 4 and chapter 5, or chapter 5 and chapter 6.

These, then, are the elements of our logical "machine" — we start with the fundamental notion of logical validity; then there are formal representations of ordinary reasonings, along with semantic validity, and validity for our two derivation systems. These elements are developed in this part. In later parts we turn to thinking about how these parts work. In particular, we begin thinking *how* to reason about logic (Part II), *demonstrate* that the same arguments come out valid by semantic methods as come out valid by the derivation methods (Part III), and develop application of the methods to arithmetic and computation (Part IV). But first we have to say what the elements are. And that is the task we set ourselves in this part.

Chapter 1

Logical Validity and Soundness

Symbolic logic is a tool or machine for the identification of argument goodness. It makes sense to begin, however, not with the machine, but by saying something about this argument goodness that the machinery is supposed to identify. That is the task of this chapter.

But first, we need to say what an argument is.

AR An *argument* is some sentences, one of which (the *conclusion*) is taken to be supported by the remaining sentences (the *premises*).

> Important definitions are often offset and given a short name as above. Then there may be appeal to the definition by its name, in this case, 'AR'.

So some sentences are an argument depending on whether premises are taken to support a conclusion. Such support is often indicated by words or phrases of the sort, 'so', 'it follows', 'therefore', or the like. We will typically indicate the division by a simple line between premises and conclusion. Roughly, an argument is good if premises do what they are taken to do, if they actually support the conclusion. An argument is bad if they do not accomplish what they are taken to do, if they do not actually support the conclusion.

Logical validity and soundness correspond to different ways an argument can go wrong. Consider the following two arguments:

(A)
Only citizens can vote
Hannah is a citizen
———
Hannah can vote

(B)
All citizens can vote
Hannah is a citizen
———
Hannah can vote

The line divides premises from conclusion, indicating that the premises are supposed to support the conclusion. Thus these are arguments. But these arguments go wrong in different ways. The premises of argument (A) are true; as a matter of fact, only citizens can vote, and Hannah (my daughter) is a citizen. But she cannot vote; she is not old enough. So the conclusion is false. Thus, in argument (A), the relation between the premises and the conclusion is defective. Even though the premises are true, there is no guarantee that the conclusion is true as well. We will say that this argument is *logically invalid*. In contrast, argument (B) is logically valid. If its premises were true, the conclusion would be true as well. So the *relation* between the premises and conclusion is not defective. The problem with this argument is that the premises are not true — not all citizens can vote. So argument (B) is defective, but in a different way. We will say that it is *logically unsound*.

The task of this chapter is to define and explain these notions of logical validity and soundness. I begin with some preliminary notions, then turn to official definitions of logical validity and soundness, and finally to some consequences of the definitions.

1.1 Consistent Stories

Given a certain notion of a *possible* or *consistent* story, it is easy to state definitions for logical validity and soundness. So I begin by identifying the kind of stories that matter. Then we will be in a position to state the definitions, and apply them in some simple cases.

Let us begin with the observation that there are different sorts of possibility. Consider, say, “Hannah could make it in the WNBA.” This seems true. She is reasonably athletic, and if she were to devote herself to basketball over the next few years, she might very well make it in the WNBA. But wait! Hannah is only a kid — she rarely gets the ball even to the rim from the top of the key — so there is no way she could make it in the WNBA. So she both could and could not make it. But this cannot be right! What is going on? Here is a plausible explanation: Different sorts of possibility are involved. When we hold fixed current abilities, we are inclined to say there is no way she could make it. When we hold fixed only general physical characteristics, and allow for development, it is natural to say that she might. The scope of what is possible varies with whatever constraints are in play. The weaker the constraints, the broader the range of what is possible.

The sort of possibility we are interested in is *very* broad, and constraints are correspondingly weak. We will allow that a story is *possible* or *consistent* so long as it involves no *internal* contradiction. A story is impossible when it collapses from

within. For this it may help to think about the way you respond to ordinary fiction. Consider, say, *Bill and Ted's Excellent Adventure* (set and partly filmed locally for me in San Dimas, CA). Bill and Ted travel through time in a modified phone booth collecting historical figures for a history project. Taken seriously, this is bizarre, and it is particularly outlandish to think that a *phone booth* should travel through time. But the movie does not so far contradict itself. So you go along. So far, then, so good (excellent).

But, late in the movie, Bill and Ted have a problem breaking the historical figures out of jail. So they decide today to go back in time tomorrow to set up a diversion that will go off in the present. The diversion goes off as planned, and the day is saved. Somehow, then, as often happens in these films, the past depends on the future, at the same time as the future depends on the past. This, rather than the time travel itself, generates an internal conflict. The movie makes it the case that you cannot have today apart from tomorrow, and cannot have tomorrow apart from today. Perhaps today and tomorrow have always been repeating in an eternal loop. But, according to the movie, there were times before today and after tomorrow. So the movie faces *internal* collapse. Notice: the objection does not have *anything* to do with the way things actually are — with the nature of actual phone booths and the like; it has rather to do with the way the movie hangs together internally — it makes it impossible for today to happen without tomorrow, and for tomorrow to happen without today.[1] Similarly, we want to ask whether stories hold together *internally*. If a story holds together internally, it counts for our purposes as consistent and possible. If a story does not hold together, it is not consistent or possible.

In some cases, then, stories may be consistent with things we know are true in the real world. Thus perhaps I come home, notice that Hannah is not in her room, and imagine that she is out back shooting baskets. There is nothing inconsistent about this. But stories may remain consistent though they do not fit with what we know to be true in the real world. Here are cases of phone booths traveling through time and the like. Stories become inconsistent when they collapse internally — as when today both can and cannot happen apart from tomorrow.

As with a movie or novel, we can say that different things are true or false *in* our stories. In *Bill and Ted's Excellent Adventure* it is true that Bill and Ted travel through

[1]In more consistent cases of time travel (in the movies) time seems to move in a sort of 'Z' so that after yesterday and today, there is *another* yesterday and *another* today. So time does not return to the very point at which it first turns back. In the trouble cases, however, time seems to move in a sort of "loop" so that a point on the path to today (this very day) goes through tomorrow. With this in mind, it is interesting to think about say, the *Terminator* and *Back to the Future* movies and, maybe more consistent, *Groundhog Day*. Even if I am wrong, and *Bill and Ted* is internally consistent, the overall point should be clear. And it should be clear that I am not saying anything serious about time travel.

time in a phone booth, but false that they go through time in a DeLorean (as in the *Back to the Future* films). In the real world, of course, it is false that phone booths go through time, and false that DeLoreans go through time. Officially, a complete story is always *maximal* in the sense that *any* sentence is either true or false in it. A story is *inconsistent* when it makes some sentence both true and false. Since, ordinarily, we do not describe every detail of what is true and what is false when we tell a story, what we tell is only part of a maximal story. In practice, however, it will be sufficient for us merely to give or fill in whatever details are relevant in a particular context.

But there are a couple of cases where we cannot say when sentences are true or false in a story. The first is when stories we tell do not fill in relevant details. In *The Wizard of Oz*, it is true that Dorothy wears red shoes. But neither the movie nor the book have anything to say about whether she likes Twinkies. By themselves, then, neither the book nor the movie give us enough information to tell whether "Dorothy likes Twinkies" is true or false in the story. Similarly, there is a problem when stories are inconsistent. Suppose according to some story,

(a) All dogs can fly

(b) Fido is a dog

(c) Fido cannot fly

Given (a), all dogs fly; but from (b) and (c), it seems that not all dogs fly. Given (b), Fido is a dog; but from (a) and (c) it seems that Fido is not a dog. Given (c), Fido cannot fly; but from (a) and (b) it seems that Fido can fly. The problem is not that inconsistent stories say too little, but rather that they say too much. When a story is inconsistent, we will simply refuse to say that it makes any sentence (simply) true or false.[2]

Consider some examples: (a) The true story, "Everything is as it actually is." Since no contradiction is actually true, this story involves no contradiction; so it is internally consistent and possible.

(b) "All dogs can fly: over the years, dogs have developed extraordinarily large and muscular ears; with these ears, dogs can fly." It is bizarre, but not obviously inconsistent. If we allow the consistency of stories according to which monkeys fly, as in *The Wizard of Oz*, or elephants fly, as in *Dumbo*, then we should allow that this story is consistent as well.

[2]The intuitive picture developed above should be sufficient for our purposes. However, we are on the verge of vexed issues. For further discussion, you may want to check out the vast literature on "possible worlds." Contributions of my own include the introductory article, "Modality," in *The Continuum Companion to Metaphysics*.

(c) "All dogs can fly, but my dog Fido cannot; Fido's ear was injured while he was chasing a helicopter, and he cannot fly." This is *not* internally consistent. If all dogs can fly and Fido is a dog, then Fido can fly. You might think that Fido remains a flying sort of thing. In evaluating internal consistency, however, we require that *meanings remain the same*: If "can fly" means just "is a flying sort of thing," then the story falls apart insofar as it says both that Fido is and is not that sort of thing; if "can fly" means "is himself able to fly," then the story falls apart insofar as it says that Fido himself both is and is not able to fly. So long as "can fly" means the same in each use, the story is sure to fall apart insofar as it says both that Fido is and is not that sort of thing.

(d) "Germany won WWII; the United States never entered the war; after a long and gallant struggle, England and the rest of Europe surrendered." It did not happen; but the story does not contradict itself. For our purposes, then it counts as possible.

(e) "1 + 1 = 3; the numerals '2' and '3' are switched ('1', '3', '2', '4', '5', '6', '7'. . .); so that taking one thing and one thing results in three things." This story does not hang together. Of course numerals can be switched; but switching numerals does not make one thing and one thing three things! We tell stories in our own language (imagine that you are describing a foreign-language film in English). According to the story, people can say correctly '1 + 1 = 3', but this does not make it the case that 1 + 1 = 3. Compare a language like English except that 'fly' means 'bark'; and consider a movie where dogs are ordinary, but people correctly assert, in this language, "dogs fly": it would be wrong to say, in *English*, that this is a movie in which *dogs fly*. And, similarly, we have not told a story where 1 + 1 = 3.

> Some authors prefer talk of "possible worlds," "possible situations" or the like to that of consistent stories. It is conceptually simpler to stick with stories, as I have, than to have situations and distinct descriptions of them. However, it is worth recognizing that our consistent stories are or describe possible situations, so that the one notion matches up directly with the others.

E1.1. Say whether each of the following stories is internally consistent or inconsistent. In either case, explain why.

*a. Smoking cigarettes greatly increases the risk of lung cancer, although most people who smoke cigarettes do not get lung cancer.

b. Joe is taller than Mary, but Mary is taller than Joe.

*c. Abortion is always morally wrong, though abortion is morally right in order to save a woman's life.

d. Mildred is Dr. Saunders's daughter, although Dr. Saunders is not Mildred's father.

*e. No rabbits are nearsighted, though some rabbits wear glasses.

f. Ray got an 'A' on the final exam in both Phil 200 and Phil 192. But he got a 'C' on the final exam in Phil 192.

*g. Bill Clinton was never president of the United States, although Hillary is president right now.

h. Egypt, with about 100 million people is the most populous country in Africa, and Africa contains the most populous country in the world. But the United States has over 200 million people.

*i. The death star is a weapon more powerful than that in any galaxy, though there is, in a galaxy far far away, a weapon more powerful than it.

j. Luke and the rebellion valiantly battled the evil empire, only to be defeated. The story ends there.

E1.2. For each of the following sentences, (i) say whether it is true or false in the real world and then (ii) say if you can whether it is true or false according to the accompanying story. In each case, explain your answers. Do not forget about contexts where we refuse to say sentences are true or false. The first problem is worked as an example.

a. Sentence: Aaron Burr was never a president of the United States.

Story: Aaron Burr was the first president of the United States, however he turned traitor and was impeached and then executed.

(i) It is *true* in the real world that Aaron Burr was never a president of the United States. (ii) But the story makes the sentence *false*, since the story says Burr was the first president.

b. Sentence: In 2006, there were still buffalo.

Story: A thundering herd of buffalo overran Phoenix Arizona in early 2006. The city no longer exists.

*c. Sentence: After overrunning Phoenix in early 2006, a herd of buffalo overran Newark, New Jersey.

Story: A thundering herd of buffalo overran Phoenix Arizona in early 2006. The city no longer exists.

d. Sentence: There has been an all-out nuclear war.

Story: After the all-out nuclear war, John Connor organized resistance against the machines — who had taken over the world for themselves.

*e. Sentence: Jack Nicholson has swum the Atlantic.

Story: No human being has swum the Atlantic. Jack Nicholson and Bill Clinton and you are all human beings, and at least one of you swam all the way across!

f. Sentence: Some people have died as a result of nuclear explosions.

Story: As a result of a nuclear blast that wiped out most of this continent, you have been dead for over a year.

*g. Sentence: Your instructor is not a human being.

Story: No beings from other planets have ever made it to this country. However, your instructor made it to this country from another planet.

h. Sentence: Lassie is both a television and movie star.

Story: Dogs have super-big ears and have learned to fly. Indeed, all dogs can fly. Among the many dogs are Lassie and Rin Tin Tin.

*i. Sentence: The Yugo is the most expensive car in the world.

Story: Jaguar and Rolls Royce are expensive cars. But the Yugo is more expensive than either of them.

j. Sentence: Lassie is a bird who has learned to fly.

Story: Dogs have super-big ears and have learned to fly. Indeed, all dogs can fly. Among the many dogs are Lassie and Rin Tin Tin.

1.2 The Definitions

The definition of logical validity depends on what is true and false in consistent stories. The definition of soundness builds directly on the definition of validity. Note:

in offering these definitions, I *stipulate* the way the terms are to be used; there is no attempt to say how they are used in ordinary conversation; rather, we say what they will mean for us in this context.

LV An argument is *logically valid* if and only if (iff) there is no consistent story in which all the premises are true and the conclusion is false.

LS An argument is *logically sound* iff it is logically valid and all of its premises are true in the real world.

Logical (deductive) validity and soundness are to be distinguished from *inductive* validity and soundness or success. For the inductive case, it is natural to focus on the *plausibility* or the *probability* of stories — where an argument is relatively strong when stories that make the premises true and conclusion false are relatively implausible. Logical (deductive) validity and soundness are thus a sort of limiting case, where stories that make premises true and conclusion false are not merely implausible, but impossible. In a deductive argument, conclusions are supposed to be *guaranteed*; in an inductive argument, conclusions are merely supposed to be made probable or plausible. For mathematical logic, we set the inductive case to the side, and focus on the deductive.

1.2.1 Invalidity

It will be easiest to begin thinking about *invalidity*. If an argument is logically valid, there is no consistent story that makes the premises true and conclusion false. So, to show that an argument is invalid, it is enough to *produce* even one consistent story that makes premises true and conclusion false. Perhaps there are stories that result in other combinations of true and false for the premises and conclusion; this does not matter for the definition. However, if there is even one story that makes premises true and conclusion false then, by definition, the argument is not logically valid — and if it is not valid, by definition, it is not logically sound. We can work through this reasoning by means of a simple *invalidity test*. Given an argument, this test has the following four stages.

IT
a. List the premises and negation of the conclusion.
b. Produce a consistent story in which the statements from (a) are all true.
c. Apply the definition of validity.
d. Apply the definition of soundness.

We begin by considering what needs to be done to show invalidity. Then we do it. Finally we apply the definitions to get the results. For a simple example, consider the following argument,

(C)
Eating Brussels sprouts results in good health
Ophilia has good health
———
Ophilia has been eating brussels sprouts

The definition of validity has to do with whether there are consistent stories in which the premises are true and the conclusion false. Thus, in the first stage, we simply write down what would be the case in a story of this sort.

a. List premises and negation of conclusion.	In any story with the premises true and conclusion false, (1) Eating brussels sprouts results in good health (2) Ophilia has good health (3) Ophilia has not been eating brussels sprouts

Observe that the conclusion is reversed! At this stage we are not giving an argument. We rather merely list what is the case when the premises are true and conclusion false. Thus there is no line between premises and the last sentence, insofar as there is no suggestion of support. It is easy enough to repeat the premises. Then we say what is required for the conclusion to be *false*. Thus, "Ophilia has been eating brussels sprouts" is false if Ophilia has not been eating brussels sprouts. I return to this point below, but that is enough for now.

An argument is invalid if there is even one consistent story that makes the premises true and the conclusion false. Thus, to show invalidity, it is enough to *produce* a consistent story that makes the premises true and conclusion false.

b. Produce a consistent story in which the statements from (a) are all true.	Story: Eating brussels sprouts results in good health, but eating spinach does so as well; Ophilia is in good health but has been eating spinach, not brussels sprouts.

For the statements listed in (a): we satisfy (1) insofar as eating brussels sprouts results in good health; (2) is satisfied since Ophilia is in good health; and (3) is satisfied since Ophilia has not been eating brussels sprouts. The story *explains* how she manages to maintain her health without eating brussels sprouts, and so the consistency of (1) - (3) together. The story does not have to be true — and, of course, many different stories

will do. All that matters is that there is a *consistent* story in which the premises of the original argument are true, and the conclusion is false.

Producing a story that makes the premises true and conclusion false is the creative part. What remains is to apply the definitions of validity and soundness. By LV an argument is logically valid only if there is no consistent story in which the premises are true and the conclusion is false. So if, as we have demonstrated, there is such a story, the argument cannot be logically valid.

c. Apply the definition of validity.	This is a consistent story that makes the premises true and the conclusion false; thus, by definition, the argument is not logically valid.

By LS, for an argument to be sound, it must have its premises true in the real world *and* be logically valid. Thus if an argument fails to be logically valid, it automatically fails to be logically sound.

d. Apply the definition of soundness.	Since the argument is not logically valid, by definition, it is not logically sound.

Given an argument, the definition of validity depends on stories that make the premises true and the conclusion false. Thus, in step (a) we simply list claims required of any such story. To show invalidity, in step (b), we produce a consistent story that satisfies each of those claims. Then in steps (c) and (d) we apply the definitions to get the final results; for invalidity, these last steps are the same in every case.

It may be helpful to think of stories as a sort of "wedge" to pry the premises of an argument off its conclusion. We pry the premises off the conclusion if there is a consistent way to make the premises true and the conclusion not. If it is possible to insert such a wedge between the premises and conclusion, then a defect is exposed in the way premises are connected to the conclusion. Observe that this is just what we did with argument (A) at the beginning of the chapter: Faced with the premises that only citizens can vote and Hannah is a citizen, it was natural to worry that she might be under-age and so cannot vote. But this is precisely to produce a story that makes the premises true and conclusion false. Thus our method is not "strange" or "foreign"! Rather, it makes rigorous what has seemed natural from the start. Observe also that the flexibility allowed in consistent stories (with flying dogs and the like) corresponds directly to the strength of connections required. If connections are sufficient to resist all such attempts to wedge the premises off the conclusion, they are significant indeed.

Here is another example of our method. Though the argument may seem on its face not to be a very good one, we can expose its failure by our methods — in fact, again, our method may formalize or make rigorous a way you very naturally think about cases of this sort. Here is the argument,

(D)
I shall run for president
———
I will be one of the most powerful men on earth

To show that the argument is invalid, we turn to our standard procedure.

a. In any story with the premise true and conclusion false,

   1. I shall run for president
   2. I will not be one of the most powerful men on earth

b. Story: I do run for president, but get no financing and gain no votes; I lose the election. In the process, I lose my job as a professor and end up begging for scraps outside a Domino's Pizza restaurant. I fail to become one of the most powerful men on earth.

c. This is a consistent story that makes the premise true and the conclusion false; thus, by definition, the argument is not logically valid.

d. Since the argument is not logically valid, by definition, it is not logically sound.

This story forces a wedge between the premise and the conclusion. Thus we use the definition of validity to explain why the conclusion does not properly follow from the premises. It is, perhaps, obvious that *running* for president is not enough to make me one of the most powerful men on earth. Our method forces us to be very explicit about why: running for president leaves open the option of losing, so that the premise does not force the conclusion. Once you get used to it, then, our method may come to seem a natural approach to arguments.

If you follow this method for showing invalidity, the place where you are most likely to go wrong is stage (b), telling stories where the premises are true and the conclusion false. Be sure that your story is consistent, and that it verifies *each* of the claims from stage (a). If you do this, you will be fine.

E1.3. Use our invalidity test to show that each of the following arguments is not logically valid, and so not logically sound. Understand terms in their most natural sense.

*a. If Joe works hard, then he will get an ‘A’
Joe will get an ‘A’
———
Joe works hard

b. Harry had his heart ripped out by a government agent
———
Harry is dead

c. Everyone who loves logic is happy
Jane does not love logic
———
Jane is not happy

d. Our car will not run unless it has gasoline
Our car has gasoline
———
Our car will run

e. Only citizens can vote
Hannah is a citizen
———
Hannah can vote

1.2.2 Validity

For a given argument, if you cannot find a story that makes the premises true and conclusion false, you may begin to suspect that it is valid. However, mere failure to demonstrate invalidity does not demonstrate validity — for all we know, there might be some tricky story we have not thought of yet. So, to show validity, we need another approach. If we could show that every story which makes the premises true and conclusion false is *inconsistent*, then we could be sure that no *consistent* story makes the premises true and conclusion false — and so we could conclude that the argument is valid. Again, we can work through this by means of a procedure, this time a *validity test*.

VT
a. List the premises and negation of the conclusion.
b. Expose the inconsistency of such a story.
c. Apply the definition of validity.
d. Apply the definition of soundness.

In this case, we begin in just the same way. The key difference arises at stage (b). For an example, consider this sample argument.

(E)

No car is a person
My mother is a person
———
My mother is not a car

Since LV has to do with stories where the premises are true and the conclusion false, as before we begin by listing the premises together with the negation of the conclusion.

a. List premises and negation of conclusion.	In any story with the premises true and conclusion false, (1) No car is a person (2) My mother is a person (3) My mother is a car

Any story where "My mother is not a car" is false, is one where my mother is a car (perhaps along the lines of the much reviled 1965 TV series, "My Mother the Car.").

For invalidity, we would produce a consistent story in which (1) - (3) are all true. In this case, to show that the argument is valid, we show that this *cannot* be done. That is, we show that no story that makes each of (1) - (3) true is consistent.

b. Expose the inconsistency of such a story.	In any such story, Given (1) and (3), (4) My mother is not a person Given (2) and (4), (5) My mother is and is not a person

The reasoning should be clear if you focus *just on the specified lines*. Given (1) and (3), if no car is a person and my mother is a car, then my mother is not a person. But then my mother is a person from (2) and not a person from (4). So we have our goal: any story with (1) - (3) as members contradicts itself and therefore is not consistent. Observe that we could have reached this result in other ways. For example, we might have reasoned from (1) and (2) that (4′), my mother is not a car; and then from (3) and (4′) to the result that (5′) my mother is and is not a car. Either way, an inconsistency is exposed. Thus, as before, there are different options for this creative part.

Now we are ready to apply the definitions of logical validity and soundness. First,

c. Apply the definition of validity.	So no consistent story makes the premises true and conclusion false; so by definition, the argument is logically valid.

For the invalidity test, we produce a consistent story that "hits the target" from stage (a), to show that the argument is invalid. For the validity test, we show that any attempt to hit the target from stage (a) must collapse into inconsistency: no consistent story includes each of the elements from stage (a) so that *there is no consistent story in which the premises are true and the conclusion false*. So by application of LV the argument is logically valid.

Given that the argument is logically valid, LS makes logical soundness depend on whether the premises are true in the real world. Suppose we think the premises of our argument are in fact true. Then,

d. Apply the definition of soundness.	In the real world no car is a person and my mother is a person, so all the premises are true; so since the argument is also logically valid, by definition, it is logically sound.

Observe that LS requires for logical soundness that an argument is logically valid and that its *premises* are true in the real world. Thus we are no longer thinking about merely possible stories! And we do not say anything at this stage about claims other than the premises of the original argument! Thus we do not make any claim about the truth or falsity of the conclusion, "my mother is not a car." Rather, the observations have entirely to do with the two premises, "no car is a person" and "my mother is a person." When an argument is valid and the premises are true in the real world, by LS, it is logically sound.

But it will not always be the case that a valid argument has true premises. Say "My Mother the Car" is (surprisingly) a documentary about a person rein*car*nated as a car (the premise of the show) and therefore a true account of some car that is a person. Then some cars are persons and the first premise is false; so you would have to respond as follows,

d. Since in the real world some cars are persons, the first premise is not true. So, though the argument is logically valid, by definition it is not logically sound.

Another option is that you are in doubt about reincarnation into cars, and in particular about whether some cars are persons. In this case you might respond as follows,

d. Although in the real world my mother is a person, I cannot say whether no car is a person; so I cannot say whether the first premise is true. So though the argument is logically valid, I cannot say whether it is logically sound.

So given validity there are three options: (i) You are in a position to identify all of the premises as true in the real world. In this case, you should do so, and apply the definition for the conclusion that the argument is logically sound. (ii) You are in a position to say that at least one of the premises is false in the real world. In this case, you should do so, and apply the definition for the conclusion that the argument is not logically sound. (iii) You cannot identify any premise as false, but neither can you identify them all as true. In this case, you should explain the situation and apply the definition for the result that you are not in a position to say whether the argument is logically sound.

Again, given an argument we say in step (a) what would be the case in any story that makes the premises true and the conclusion false. Then, at step (b), instead of finding a consistent story in which the premises are true and conclusion false, we show that there is no such thing. Steps (c) and (d) apply the definitions for the final results. Observe that only one method can be correctly applied in a given case! If we can produce a consistent story according to which the premises are true and the conclusion is false, then it is not the case that no consistent story makes the premises true and the conclusion false. Similarly, if no consistent story makes the premises true and the conclusion false, then we will not be able to produce a consistent story that makes the premises true and the conclusion false.

In this case, the most difficult steps are (a) and (b), where we say what is the case in every story that makes the premises true and the conclusion false. For an example, consider the following argument.

(F)
Some collies can fly
All collies are dogs
_____
All dogs can fly

It is invalid. We can easily tell a story that makes the premises true and the conclusion false — say one where Lassie is a collie who can fly, but otherwise things are as usual. Suppose, however, that we proceed with the validity test as follows,

a. In any story with the premises true and conclusion false,

   (1) Some collies can fly
   (2) All collies are dogs
   (3) No dogs can fly

b. In any such story,

Given (1) and (2),
(4) Some dogs can fly
Given (3) and (4),
(5) Some dogs can and cannot fly

c. So no consistent story makes the premises true and conclusion false; so by definition, the argument is logically valid.

d. Since in the real world no collies can fly, the first premise is not true. So, though the argument is logically valid, by definition it is not logically sound.

The reasoning at (b), (c) and (d) is correct. Any story with (1) - (3) is inconsistent. But something is wrong. (Can you see what?) There is a mistake at (a): It is not the case that every story that makes the premises true and conclusion false makes (3) true. The negation of "All dogs can fly" is not "No dogs can fly," but rather, "Not all dogs can fly" ("Some dogs cannot fly"). All it takes to falsify the claim that all dogs fly, is one dog that does not. Thus, for example, all it takes to falsify the claim that everyone will get an 'A' is one person who does not (on this, see the extended discussion on p. 20). We have indeed shown that every story of a certain sort is inconsistent, but have not shown that every story which makes the premises true and conclusion false is inconsistent. In fact, as we have seen, there are consistent stories that make the premises true and conclusion false. Similarly, in step (b) it is easy to get confused if you consider too much information at once. Ordinarily, if you focus on sentences singly or in pairs, it will be clear what must be the case in every story including those sentences. It does not matter which sentences you consider in what order, so long as you reach a contradiction, according to which something is and is not so, in the end.

So far, we have seen our procedures applied in contexts where it is given ahead of time whether an argument is valid or invalid. And some exercises have been this way too. But not all situations are so simple. In the ordinary case, it is not given whether an argument is valid or invalid. In this case, there is no magic way to say ahead of time which of our two tests, IT or VT applies. The only thing to do is to try one way — if it works, fine. If it does not, try the other. It is perhaps most natural to begin by looking for stories to pry the premises off the conclusion. If you can find a consistent story to make the premises true and conclusion false, the argument is invalid. If you cannot find any such story, you may begin to suspect that the argument is valid. This suspicion does not itself amount to a demonstration of validity! But you might try to turn your suspicion into such a demonstration by attempting the validity method. Again, if one procedure works, the other better not!

Negation and Quantity

In general you want to be careful about negations. To negate any claim $\mathcal{P}$ it is always correct to write simply, *it is not the case that* $\mathcal{P}$. You may choose to do this for conclusions in the first step of our procedures. At some stage, however, you will need to understand what the negation comes to. We have chosen to offer interpreted versions in the text. It is easy enough to see that,

My mother is a car and My mother is not a car

negate one another. However, there are cases where caution is required. This is particularly the case where quantity terms are involved.

In the first step of our procedures, we say what is the case in *any* story where the premises are true and the conclusion is false. The negation of a claim states what is *required* for falsity, and so meets this condition. If I say there are at least ten apples in the basket, my claim is of course false if there are only three. But not every story where my claim is false is one in which there are three apples. Rather, my claim is false just in case there are less than ten. *Any* story in which there are less than ten makes my claim false.

A related problem arises with other quantity terms. To bring this out, consider grade examples: First, if a professor says, "everyone will not get an 'A'," she says something disastrous. To deny it, all you need is one person to get an 'A'. In contrast, if she says, "someone will not get an 'A'" ("not everyone will get an 'A'"), she says only what you expect from the start. To deny it, you need that everyone will get an 'A'. Thus the following pairs negate one another.

Everybody will get an 'A' and Somebody will not get an 'A'
Somebody will get an 'A' and Everybody will not get an 'A'

A sort of rule is that pushing or pulling 'not' past 'all' or 'some' flips one to the other. But it is difficult to make rules for arbitrary quantity terms. So it is best just to think about what you are saying, perhaps with reference to examples like these. Thus the following also are negations of one another.

Somebody will get an 'A' and Nobody will get an 'A'
Only jocks will get an 'A' and Some non-jock will get an 'A'

The first works because "nobody will get an 'A'" is just like "everybody will not get an 'A'," so the first pair reduces to the parallel one above. In the second case, everything turns on whether a non-jock gets an 'A': if none does, then only jocks will get an 'A'; if one or more do, then some non-jock does get an 'A'.

E1.4. Use our validity procedure to show that each of the following is logically valid, and to decide (if you can) whether it is logically sound.

*a. If Bill is president, then Hillary is first lady
Hillary is not first lady
———
Bill is not president

b. Only fools find love
Elvis was no fool
———
Elvis did not find love

c. If there is a good and omnipotent god, then there is no evil
There is evil
———
There is no good and omnipotent god

d. All sparrows are birds
All birds fly
———
All sparrows fly

e. All citizens can vote
Hannah is a citizen
———
Hannah can vote

E1.5. Use our procedures to say whether the following are logically valid or invalid, and sound or unsound. Hint: You may have to do some experimenting to decide whether the arguments are logically valid or invalid — and so to decide which procedure applies.

a. If Bill is president, then Hillary is first lady
Bill is president
———
Hillary is first lady

b. Most professors are insane
TR is a professor
———
TR is insane

*c. Some dogs have red hair
Some dogs have long hair
———
Some dogs have long red hair

d. If you do not strike the match, then it does not light
The match lights
————
You strike the match

e. Shaq is taller than Kobe
Kobe is at least as tall as TR
————
Kobe is taller than TR

1.3 Some Consequences

We now know what logical validity and soundness are and should be able to identify them in simple cases. Still, it is one thing to know what validity and soundness are, and another to know how we can use them. So in this section I turn to some consequences of the definitions.

1.3.1 Soundness and Truth

First, a consequence we want: The conclusion of every sound argument is true in the real world. Observe that this is *not* part of what we require to show that an argument is sound. LS requires just that an argument is valid and that its *premises* are true. However, it is a consequence of these requirements that the conclusion is true as well. To see this, suppose we have a sound two-premise argument, and think about the nature of the true story. The premises and conclusion must fall into one of the following combinations of true and false in the real world:

1	2	3	4	5	6	7	8
T	T	T	F	T	F	F	F
T	T	F	T	F	T	F	F
T	F	T	T	F	F	T	F

If the argument is logically sound, it is logically valid; so no consistent story makes the premises true and the conclusion false. But the true story is a consistent story. So we can be sure that the true story does not result in combination (2). So far, the true story might fall into any of the other combinations. Thus the conclusion of a valid argument may or may not be true in the real world. But if an argument is sound, its premises are true in the real world. So, for a sound argument, we can be sure that the premises do not fall into any of the combinations (3) - (8). (1) is the only combination left: in the true story, the conclusion is true. And, in general, if an argument is sound, its conclusion is true in the real world: If there is no consistent

story where the premises are true and the conclusion is false, and the premises are in fact true, then the conclusion must be true as well. Or with true premises, if the conclusion were false then the real world would correspond to a story with premises true and conclusion false, and the argument would not be valid after all — so the argument would not be sound. Note again: we do not need that the conclusion is true in the real world in order to say that an argument is sound, and saying that the conclusion is true is no part of our procedure for validity or soundness! Rather, by discovering that an argument is logically valid and that its *premises* are true, we *establish* that it is sound; this gives us the result that its conclusion therefore is true. And that is just what we want.

1.3.2 Validity and Form

Some of the arguments we have seen so far are of the same general *form*. Thus both of the arguments on the left have the form on the right.

(G)	If Joe works hard, then he will get an 'A' Joe works hard ——— Joe will get an 'A'	If Hannah is a citizen then she can vote Hannah is a citizen ——— Hannah can vote	If $\mathscr{P}$ then $\mathscr{Q}$ $\mathscr{P}$ ——— $\mathscr{Q}$

As it turns out, all arguments of this form are valid. In contrast, the following arguments with the indicated form are not.

(H)	If Joe works hard then he will get an 'A' Joe will get an 'A' ——— Joe works hard	If Hannah can vote, then she is a citizen Hannah is a citizen ——— Hannah can vote	If $\mathscr{P}$ then $\mathscr{Q}$ $\mathscr{Q}$ ——— $\mathscr{P}$

There are stories where, say, Joe cheats for the 'A', or Hannah is a citizen but not old enough to vote. In these cases, there is some other way to obtain condition $\mathscr{Q}$ than by having $\mathscr{P}$ — this is what the stories bring out. And, generally, it is often possible to characterize arguments by their forms, where a form is *valid* iff every instance of it is logically valid. Thus the first form listed above is valid, and the second not. In fact, the logical machine to be developed in chapters to come takes advantage of certain very general formal or structural features of arguments to demonstrate the validity of arguments with those features.

For now, it is worth noting that some presentations of critical reasoning (which you may or may not have encountered), take advantage of such patterns, listing typical ones that are valid, and typical ones that are not (for example, Cederblom and

Paulsen, *Critical Reasoning*). A student may then identify valid and invalid arguments insofar as they match the listed forms. This approach has the advantage of simplicity — and one may go quickly to applications of the logical notions to concrete cases. But the approach is limited to application of listed forms, and so to a very limited range, whereas our definition has application to arbitrary arguments. Further, a mere listing of valid forms does not explain their relation to truth, whereas the definition is directly connected. Similarly, our logical machine develops an account of validity for arbitrary forms (within certain ranges). So we are pursuing a general account or theory of validity that goes well beyond the mere lists of these other more traditional approaches.[3]

1.3.3 Relevance

Another consequence seems less welcome. Consider the following argument.

(I)
Snow is white
Snow is not white
———
All dogs can fly

It is natural to think that the premises are not connected to the conclusion in the right way — for the premises have nothing to do with the conclusion — and that this argument therefore should not be logically valid. But if it is not valid, by definition, there is a consistent story that makes the premises true and the conclusion false. And, in this case, there is no such story, for *no consistent story makes the premises true*. Thus, by definition, this argument is logically valid. The procedure applies in a straightforward way. Thus,

a. In any story that makes the premises true and conclusion false,

   (1) Snow is white
   (2) Snow is not white
   (3) Some dogs cannot fly

b. In any such story,

[3] Some authors introduce a notion of *formal validity* (maybe in the place of logical validity as above) such that an argument is formally valid iff it has some valid form. As above, formal validity is parasitic on logical validity, together with a to-be-specified notion of form. But if an argument is formally valid, it is logically valid. So if our logical machine is adequate to identify formal validity, it identifies logical validity as well.

Given (1) and (2),
(4) Snow is and is not white

c. So no consistent story makes the premises true and conclusion false; so by definition, the argument is logically valid.

d. Since in the real world snow is white, the second premise is not true. So, though the argument is logically valid, by definition it is not logically sound.

This seems bad! Intuitively, there is something wrong with the argument. But, on our official definition, it is logically valid. One might rest content with the observation that, even though the argument is logically valid, it is not logically sound. But this does not remove the general worry. For this argument,

(J) $\dfrac{\text{There are fish in the sea}}{1 + 1 = 2}$

has all the problems of the other and is logically *sound* as well. (Why?) One might, on the basis of examples of this sort, decide to reject the (classical) account of validity with which we have been working. Some do just this.[4] But, for now, let us see what can be said in defense of the classical approach. (And the classical approach is, no doubt, the approach you have seen or will see in any standard course on critical thinking or logic.)

As a first line of defense, one might observe that the conclusion of every sound argument is true and ask, “What more do you want?” We use arguments to demonstrate the truth of conclusions. And nothing we have said suggests that sound arguments do not have true conclusions: An argument whose premises are inconsistent, is sure to be unsound. And an argument whose conclusion cannot be false, is sure to have a true conclusion. So soundness may seem sufficient for our purposes. Even though we accept that there remains something about argument goodness that soundness leaves behind, we can insist that soundness is useful as an intellectual tool. Whenever it is the truth or falsity of a conclusion that matters, we can profitably employ the classical notions.

But one might go further, and dispute even the suggestion that there is something about argument goodness that soundness leaves behind. Consider the following two argument forms.

[4]Especially the so-called “relevance” logicians. For an introduction, see Graham Priest, *Non-Classical Logics*. But his text presumes mastery of material corresponding to Part I and Part II (or at least Part I with chapter 7) of this one. So the non-classical approaches develop or build on the classical one developed here.

(ds) $\dfrac{\mathscr{P} \text{ or } \mathscr{Q}, \text{ not-}\mathscr{P}}{\mathscr{Q}}$ (add) $\dfrac{\mathscr{P}}{\mathscr{P} \text{ or } \mathscr{Q}}$

According to ds (*disjunctive syllogism*), if you are given that $\mathscr{P}$ or $\mathscr{Q}$ and that not-$\mathscr{P}$, you can conclude that $\mathscr{Q}$. If you have cake or ice cream, and you do not have cake, you have ice cream; if you are in California or New York, and you are not in California, you are in New York; and so forth. Thus ds seems hard to deny. And similarly for add (*addition*). Where 'or' means "one or the other or both," when you are given that $\mathscr{P}$, you can be sure that $\mathscr{P}$ or anything. Say you have cake, then you have cake or ice cream, cake or brussels sprouts, and so forth; if grass is green, then grass is green or pigs have wings, grass is green or dogs fly, and so forth.

Return now to our problematic argument. As we have seen, it is valid according to the classical definition LV. We get a similar result when we apply the ds and add principles.

1.	Snow is white	premise
2.	Snow is not white	premise
3.	Snow is white or all dogs can fly	from 1 and add
4.	All dogs can fly	from 2 and 3 and ds

If snow is white, then snow is white or anything. So snow is white or dogs fly. So we use line 1 with add to get line 3. But if snow is white or dogs fly, and snow is not white, then dogs fly. So we use lines 2 and 3 with ds to reach the final result. So our principles ds and add go hand-in-hand with the classical definition of validity. The argument is valid on the classical account; and with these principles, we can move from the premises to the conclusion. If we want to reject the validity of this argument, we will have to reject not only the classical notion of validity, but also one of our principles ds or add. And it is not obvious that one of the principles should go. If we decide to retain both ds and add then, seemingly, the classical definition of validity should stay as well. If we have intuitions according to which ds and add should stay, and also that the definition of validity should go, we have conflicting intuitions. Thus our intuitions might, at least, be sensibly resolved in the classical direction.

These issues are complex, and a subject for further discussion. For now, it is enough for us to treat the classical approach as a useful tool: It is useful in contexts where what we care about is whether conclusions are true. And alternate approaches to validity typically develop or modify the classical approach. So it is natural to begin where we are, with the classical account. At any rate, this discussion constitutes a sort of acid test: If you understand the validity of the "snow is white" and "fish in the

sea" arguments (I) and (J), you are doing well — you understand *how* the definition of validity works, with its results that may or may not now seem controversial. If you do not see what is going on in those cases, then you have not yet understood how the definitions work and should return to section 1.2 with these cases in mind.

E1.6. Use our procedures to say whether the following are logically valid or invalid, and sound or unsound. Hint: You may have to do some experimenting to decide whether the arguments are logically valid or invalid — and so to decide which procedure applies.

a. Bob is over six feet tall
Bob is under six feet tall
———
Bob is disfigured

b. Marilyn is not over six feet tall
Marilyn is not under six feet tall
———
Marilyn is beautiful

*c. The earth is (approximately) round
———
There is no round square

d. There are fish in the sea
There are birds in the sky
There are bats in the belfry
———
Two dogs are more than one

e. All dogs can fly
Fido is a dog
Fido cannot fly
———
I am blessed

E1.7. Respond to each of the following.

a. Create another argument of the same form as the first set of examples (G) from section 1.3.2, and then use our regular procedures to decide whether it is logically valid and sound. Is the result what you expect? Explain.

b. Create another argument of the same form as the second set of examples (H) from section 1.3.2, and then use our regular procedures to decide whether it is logically valid and sound. Is the result what you expect? Explain.

E1.8. Which of the following are true, and which are false? In each case, explain your answers, with reference to the relevant definitions. The first is worked as an example.

a. A logically valid argument is always logically sound.

False. An argument is sound iff it is logically valid and all of its premises are true in the real world. Thus an argument might be valid but fail to be sound if one or more of its premises is false in the real world.

b. A logically sound argument is always logically valid.

*c. If the conclusion of an argument is true in the real world, then the argument must be logically valid.

d. If the premises and conclusion of an argument are true in the real world, then the argument must be logically sound.

*e. If a premise of an argument is false in the real world, then the argument cannot be logically valid.

f. If an argument is logically valid, then its conclusion is true in the real world.

*g. If an argument is logically sound, then its conclusion is true in the real world.

h. If an argument has contradictory premises (its premises are true in no consistent story), then it cannot be logically valid.

*i. If the conclusion of an argument cannot be false (is false in no consistent story), then the argument is logically valid.

j. The premises of every logically valid argument are relevant to its conclusion.

E1.9. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Logical validity

b. Logical soundness

E1.10. Do you think we should accept the classical account of validity? In an essay of about two pages, explain your position, with special reference to difficulties raised in section 1.3.3.

Chapter 2

Formal Languages

In the picture of symbolic logic from p. 2, we suggested that symbolic logic is introduced as a machine or tool to identify validity and soundness. This machine begins with formal representations of ordinary reasonings.

There are different ways to introduce a formal language. It is natural to introduce expressions of a new language in relation to expressions of one that is already familiar. Thus, a standard course in a foreign language is likely to present vocabulary lists of the sort,

chou:	cabbage
petit:	small
⋮	

But such lists do not *define* the terms of one language relative to another. It is not a legitimate criticism of a Frenchman who refers to his sweetheart as *mon petit chou* to observe that she is no cabbage. Rather, French has conventions such that sometimes '*chou*' corresponds to 'cabbage' and sometimes it does not. It is possible to use such correlations to introduce conventions of a new language. But it is also possible to introduce a language "as itself" — the way a native speaker learns it. In this case, one avoids the danger of importing conventions and patterns from one language onto the other. Similarly, the expressions of a formal language might be introduced in correlation with expressions of, say, English. But this runs the risk of obscuring just what the official definitions accomplish. Since we will be concerned extensively with what follows from the definitions, it is best to introduce our languages in their "pure" forms.

In this chapter, we develop the *grammar* of our formal languages. As a computer can check the spelling and grammar of English without reference to meaning, so we

can introduce the vocabulary and grammar of our formal languages without reference to what their expressions mean or what makes them true. We will give some hints for the way formal expressions match up with ordinary language. But do not take these as defining the formal language. The formal language has definitions of its own. And the grammar, taken alone, is completely straightforward. Taken this way, we work directly from the definitions, without "pollution" from associations with English or whatever.

2.1 Sentential Languages

Let us begin with some of those hints at least to suggest the way things will work. Consider some simple sentences of an ordinary language, say, 'Bill is happy' and 'Hillary is happy'. It will be convenient to use capital letters to abbreviate these, say, B and H. Such sentences may combine to form ones that are more complex as, '*It is not the case that* Bill is happy' or '*If* Bill is happy, *then* Hillary is happy. We shall find it convenient to express these, '$\sim$Bill is happy' and 'Bill is happy $\rightarrow$ Hillary is happy', with operators $\sim$ and $\rightarrow$. Putting these together we get, $\sim B$ and $B \rightarrow H$. Operators may be combined in obvious ways so that $B \rightarrow \sim H$ says that if Bill is happy, then Hillary is not. And so forth. We shall see that incredibly complex expressions of this sort are possible!

In the above case, simple sentences, 'Bill is happy' and 'Hillary is happy' are "atoms" and complex sentences are built out of them. This is characteristic of the *sentential* languages to be considered in this section. For the *quantificational* languages of section 2.2, certain sentence *parts* are taken as atoms. So quantificational languages expose structure beyond that considered here. However, this should be enough to give you a glimpse of the overall strategy and aims for the sentential languages of which we are about to introduce the grammar.

Specification of the grammar for a formal language breaks into specification of the *vocabulary* or symbols of the language, and specification of those expressions which count as grammatical sentences. After introducing the vocabulary, and then the grammar for our languages, we conclude with some discussion of abbreviations for official expressions.

2.1.1 Vocabulary

The specification of a formal language begins with specification of its vocabulary. In the sentential case, this includes,

VC (p) Punctuation symbols: ( )

(o) Operator symbols: $\sim$ $\rightarrow$

(s) A non-empty countable collection of sentence letters

And that is all. $\sim$ is *tilde* and $\rightarrow$ is *arrow*. Sometimes sentential languages include operators in addition to $\sim$ and $\rightarrow$ (for example, $\vee$, $\wedge$, $\leftrightarrow$).[1] Such symbols will be introduced in due time — but as abbreviations for complex official expressions. A "stripped-down" vocabulary is sufficient to accomplish what can be done with expanded ones. And when we turn to reasoning about the language and logic, it will be convenient to have simple specifications, with a stripped-down vocabulary.

> Some definitions have both a sentential and then an extended quantificational version. In this case, I adopt the convention of naming the initial sentential version in small caps. Thus the definition above is VC, and the parallel definition of the next section, VC.

In order to fully specify the vocabulary of any particular sentential language, we need to specify its sentence letters — so far as definition VC goes, different languages may differ in their collections of sentence letters. The only constraint on such specifications is that the collections of sentence letters be non-empty and countable. A collection is *non-empty* iff it has at least one member. So any sentential language has at least one sentence letter. A collection is *countable* iff its members can be correlated one-to-one with some or all of the integers. Thus, for some language, we might let the sentence letters be $A, B \ldots Z$, where these correlate with the integers $1 \ldots 26$. Or we might let there be infinitely many sentence letters, $S_0, S_1, S_2 \ldots$ where the letters are correlated by their subscripts.

Let us introduce a standard language $\mathcal{L}_s$ whose sentence letters are Roman italics $A \ldots Z$ with or without integer subscripts. Thus,

$$A \quad C \quad L_2 \quad R_3 \quad Z_{25}$$

are all sentence letters of $\mathcal{L}_s$. We will not use the subscripts very often. But they guarantee that we never run out of sentence letters! Perhaps surprisingly, as described on p. 33 and then E2.2, these letters too can be correlated with the integers. Official sentences of $\mathcal{L}_s$ are built out of this vocabulary.

To proceed, we need some conventions for talking *about* expressions of a language like $\mathcal{L}_s$. For any formal object language $\mathcal{L}$, an *expression* is a sequence of one

[1] And sometimes sentential languages are introduced with different symbols, for example, $\neg$ for $\sim$, $\supset$ for $\rightarrow$, or & for $\wedge$. It should be easy to convert between presentations of the different sorts.

Countability

To see the full range of languages which are allowed under VC, observe how multiple infinite series of sentence letters may satisfy the countability constraint. Thus, for example, suppose we have two series of sentence letters, $A_0, A_1 \ldots$ and $B_0, B_1 \ldots$ These can be correlated with the integers as follows,

$$\begin{array}{ccccccc} A_0 & B_0 & A_1 & B_1 & A_2 & B_2 & \\ | & | & | & | & | & | & \ldots \\ 0 & 1 & 2 & 3 & 4 & 5 & \end{array}$$

For any integer n, A_n is matched with $2n$, and B_n with $2n + 1$. So each sentence letter is matched with some integer; so the sentence letters are countable. If there are three series, they may be correlated,

$$\begin{array}{ccccccc} A_0 & B_0 & C_0 & A_1 & B_1 & C_1 & \\ | & | & | & | & | & | & \ldots \\ 0 & 1 & 2 & 3 & 4 & 5 & \end{array}$$

so that every sentence letter is matched to some integer. And similarly for any finite number of series. And there might be 26 such series, as for our language $\mathcal{L}_s$.

In fact even this is not the most general case. If there are *infinitely* many series of sentence letters, we can still line them up and correlate them with the integers. Here is one way to proceed. Order the letters as follows,

$$\begin{array}{ccccccccc} A_0 & \rightarrow & A_1 & & A_2 & \rightarrow & A_3 & \ldots \\ & \swarrow & & \nearrow & & \swarrow & & \\ B_0 & & B_1 & & B_2 & & B_3 & \ldots \\ \downarrow & \nearrow & & \swarrow & & & & \\ C_0 & & C_1 & & C_2 & & C_3 & \ldots \\ & \swarrow & & & & & & \\ D_0 & & D_1 & & D_2 & & D_3 & \ldots \\ \vdots & & & & & & & \end{array}$$

And following the arrows, match them accordingly with the integers,

$$\begin{array}{ccccccc} A_0 & A_1 & B_0 & C_0 & B_1 & A_2 & \\ | & | & | & | & | & | & \ldots \\ 0 & 1 & 2 & 3 & 4 & 5 & \end{array}$$

so that, again, any sentence letter is matched with some integer. It may seem odd that we can line symbols up like this, but it is hard to dispute that we have done so. Thus we may say that VC is compatible with a wide variety of specifications, but also that all legitimate specifications have something in common: If a collection is countable, it is possible to sort its members into a series with a first member, a second member, and so forth.

or more elements of its vocabulary. The sentences of any language $\mathscr{L}$ are a subset of its expressions. Thus, already, it is clear that $(A \star B)$ is not an official sentence of $\mathscr{L}_{s}$. (Why?). We shall use script characters $\mathscr{A} \dots \mathscr{Z}$ to represent expressions. Insofar as these script characters are symbols for symbols, they are "metasymbols" and so part of a *metalanguage*. '$\sim$', '$\to$', '(', and ')' represent themselves. Concatenated or joined symbols in the metalanguage represent the concatenation of the symbols they represent. Thus, where $\mathscr{S}$ represents an arbitrary sentence letter, $\sim\mathscr{S}$ may represent any of, $\sim A$, $\sim B$, or $\sim Z_{24}$. But $\sim(A \to B)$ is not of that form, for it does not consist of a tilde followed by a sentence letter. However, where $\mathscr{P}$ is allowed to represent any arbitrary expression, $\sim(A \to B)$ is of the form $\sim\mathscr{P}$, for it consists of a tilde followed by an expression of some sort.

It is convenient to think of metalinguistic expressions as "mapping" onto object-language ones. Thus, with $\mathscr{S}$ restricted to sentence letters, there is a straightforward map from $\sim\mathscr{S}$ onto $\sim A$, $\sim B$, or $\sim Z_{24}$, but not from $\sim\mathscr{S}$ onto $\sim(A \to B)$.

(A)
$$\begin{array}{cccc} \sim\mathscr{S} & \sim\mathscr{S} & \sim\mathscr{S} & \sim\mathscr{S} \\ \downarrow\,\downarrow & \downarrow\,\downarrow & \downarrow\,\downarrow & \downarrow \qquad \\ \sim A & \sim B & \sim Z_{24} & \sim\overbrace{(A \to B)}^{?} \end{array}$$

In the first three cases, $\sim$ maps to itself, and $\mathscr{S}$ to a sentence letter. In the last case there is no map. We might try mapping $\mathscr{S}$ to A or B; but this would leave the rest of the expression unmatched. An object-language expression has some metalinguistic form just when there is a complete map from the metalinguistic form to it.

Say $\mathscr{P}$ may represent any arbitrary expression. Then by similar reasoning, $(A \to B) \to (A \to B)$ is of the form $\mathscr{P} \to \mathscr{P}$.

(B)
$$\begin{array}{ccc} & \mathscr{P} \to \mathscr{P} & \\ \swarrow & \downarrow & \searrow \\ \overbrace{(A \to B)} & \to & \overbrace{(A \to B)} \end{array}$$

In this case, $\mathscr{P}$ maps to all of $(A \to B)$ and $\to$ to itself. A constraint on our maps is that the use of the metavariables $\mathscr{A} \dots \mathscr{Z}$ must be consistent within a given map. Thus $(A \to B) \to (B \to B)$ is not of the form $\mathscr{P} \to \mathscr{P}$.

(C)
$$\begin{array}{ccc} & \mathscr{P} \to \mathscr{P} & \\ \swarrow & \downarrow & \\ \overbrace{(A \to B)} & \to & \overbrace{(B \to B)}^{?} \end{array} \qquad \text{or} \qquad \begin{array}{ccc} & \mathscr{P} \to \mathscr{P} & \\ & \downarrow & \searrow \\ \overbrace{(A \to B)}^{?} & \to & \overbrace{(B \to B)} \end{array}$$

We are free to associate $\mathcal{P}$ with whatever we want. However, within a given map, once $\mathcal{P}$ is associated with some expression, we have to use it consistently within that map.

Observe again that $\sim\mathcal{S}$ and $\mathcal{P} \rightarrow \mathcal{P}$ are not expressions of $\mathcal{L}_s$. Rather, we use them to talk about expressions of $\mathcal{L}_s$. And it is important to see how we can use the metalanguage to make claims about a range of expressions all at once. Given that $\sim A$, $\sim B$ and $\sim Z_{24}$ are all of the form $\sim\mathcal{S}$, when we make some claim about expressions of the form $\sim\mathcal{S}$, we say something about each of them — but not about $\sim(A \rightarrow B)$. Similarly, if we make some claim about expressions of the form $\mathcal{P} \rightarrow \mathcal{P}$, we say something with application to *ranges* of expressions. In the next section, for the specification of *formulas*, we use the metalanguage in just this way.

E2.1. Assuming that $\mathcal{S}$ may represent any sentence letter, and $\mathcal{P}$ any arbitrary expression of $\mathcal{L}_s$, use maps to determine whether each of the following expressions is (i) of the form $(\mathcal{S} \rightarrow \sim\mathcal{P})$ and then (ii) whether it is of the form $(\mathcal{P} \rightarrow \sim\mathcal{P})$. In each case, explain your answers.

a. $(A \rightarrow \sim A)$

b. $(A \rightarrow \sim(R \rightarrow \sim Z))$

c. $(\sim A \rightarrow \sim(R \rightarrow \sim Z))$

d. $((R \rightarrow \sim Z) \rightarrow \sim(R \rightarrow \sim Z))$

*e. $((\rightarrow \sim) \rightarrow \sim(\rightarrow \sim))$

E2.2. On the pattern of examples from the countability guide on p. 33, show that the sentence letters of $\mathcal{L}_s$ are countable — that is, that they can be correlated with the integers. On the scheme you produce, what integers correlate with A, B_1 and C_{10}? Hint: Supposing that A without subscript is like A_0, for any integer n, you should be able to produce a formula for the position of any A_n, and similarly for B_n, C_n and the like. Then it will be easy to find the position of any letter, even if the question is about, say, L_{125}.

2.1.2 Formulas

We are now in a position to say which expressions of a sentential language are its grammatical *formulas* and *sentences*. The specification itself is easy. We will spend a bit more time explaining how it works. For a given sentential language $\mathcal{L}$,

FR (s) If $\mathcal{S}$ is a sentence letter, then $\mathcal{S}$ is a *formula*.

($\sim$) If $\mathcal{P}$ is a formula, then $\sim\mathcal{P}$ is a *formula*.

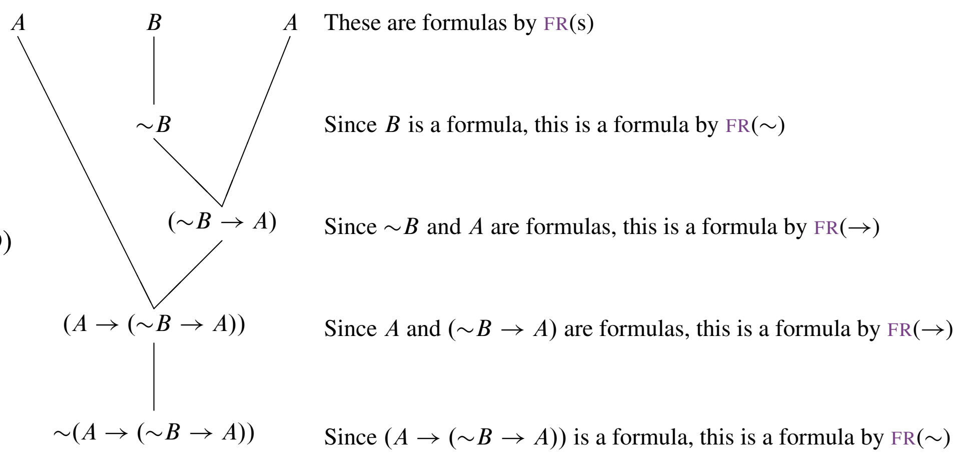
($\rightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \rightarrow \mathcal{Q})$ is a *formula*.

(CL) Any formula may be formed by repeated application of these rules.

In the quantificational case, we will distinguish a class of expressions that are formulas from those that are sentences. But, here, we simply identify the two: an expression is a *sentence* iff it is a formula.

FR is a first example of a *recursive* definition. Such definitions always build from the parts to the whole. Frequently we can use "tree" diagrams to see how they work. Thus, for example, by repeated applications of the definition, $\sim(A \rightarrow (\sim B \rightarrow A))$ is a formula and sentence of $\mathcal{L}_s$.

(D)



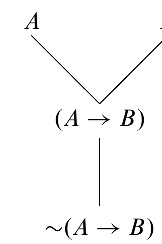
By FR(s), the sentence letters, A, B and A are formulas; given this, clauses FR($\sim$) and FR($\rightarrow$) let us conclude that other, more complex, expressions are formulas as well. Notice that, in the definition, $\mathcal{P}$ and $\mathcal{Q}$ may be any expressions that are formulas: By FR($\sim$), if B is a formula, then tilde followed by it is a formula; but similarly, if $\sim B$ and A are formulas, then an opening parenthesis followed by $\sim B$, followed by $\rightarrow$ followed by A and then a closing parenthesis is a formula; and so forth as on the tree above. You should follow through each step very carefully. In contrast, $(A\sim B)$ for example, is not a formula. A is a formula and $\sim B$ is a formula; but there is no way to put them together, by the definition, without $\rightarrow$ in between.

A recursive definition always involves some "basic" starting elements, in this case, sentence letters. These occur across the top row of our tree. Other elements are constructed, by the definition, out of ones that come before. The last, *closure*,

clause tells us that any formula is built this way. To demonstrate that an expression is a formula and a sentence, it is sufficient to construct it, according to the definition, on a tree. If an expression is not a formula, there will be no way to construct it according to the rules.

Here are a couple of last examples which emphasize the point that *you must maintain and respect parentheses* in the way you construct a formula. Thus consider,

(E)



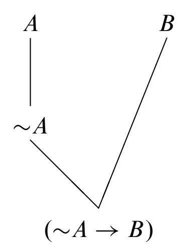
These are formulas by FR(s)

Since A and B are formulas, this is a formula by FR($\to$)

Since $(A \to B)$ is a formula, this is a formula by FR($\sim$)

And compare it with,

(F)



These are formulas by FR(s)

Since A is a formula, this is a formula by FR($\sim$)

Since $\sim A$ and B are formulas, this is a formula by FR($\to$)

Once you have $(A \to B)$ as in the first case, the only way to apply FR($\sim$) puts the tilde on the outside. To get the tilde inside the parentheses, by the rules, it has to go on first, as in the second case. The significance of this point emerges immediately below.

It will be helpful to have some additional definitions, each of which may be introduced in relation to the trees. First, for any formula $\mathcal{P}$, each formula which appears in the tree for $\mathcal{P}$ including $\mathcal{P}$ itself is a *subformula* of $\mathcal{P}$. Thus $\sim(A \to B)$ has subformulas,

$$A \qquad B \qquad (A \to B) \qquad \sim(A \to B)$$

In contrast, $(\sim A \to B)$ has subformulas,

$$A \qquad B \qquad \sim A \qquad (\sim A \to B)$$

So it matters for the subformulas how the tree is built. The *immediate* subformulas of a formula $\mathcal{P}$ are the subformulas to which $\mathcal{P}$ is directly connected by lines. Thus $\sim(A \to B)$ has one immediate subformula, $(A \to B)$; $(\sim A \to B)$ has two, $\sim A$ and

B. The *atomic* subformulas of a formula $\mathscr{P}$ are the sentence letters that appear across the top row of its tree. Thus both $\sim(A \rightarrow B)$ and $(\sim A \rightarrow B)$ have A and B as their atomic subformulas. Finally, the *main operator* of a formula $\mathscr{P}$ is the last operator added in its tree. Thus $\sim$ is the main operator of $\sim(A \rightarrow B)$, and $\rightarrow$ is the main operator of $(\sim A \rightarrow B)$. So, again, it matters how the tree is built. We sometimes speak of a formula by means of its main operator: A formula of the form $\sim\mathscr{P}$ is a *negation*; a formula of the form $(\mathscr{P} \rightarrow \mathscr{Q})$ is a *(material) conditional*, where $\mathscr{P}$ is the *antecedent* of the conditional and $\mathscr{Q}$ is the *consequent*.

Parts of a Formula

The parts of a formula are here defined in relation to its tree.

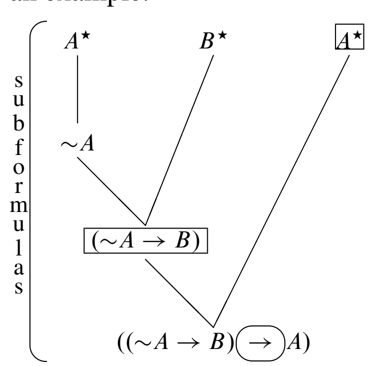
SB Each formula which appears in the tree for formula $\mathscr{P}$ including $\mathscr{P}$ itself is a *subformula* of $\mathscr{P}$.

IS The *immediate* subformulas of a formula $\mathscr{P}$ are the subformulas to which $\mathscr{P}$ is directly connected by lines.

AS The *atomic* subformulas of a formula $\mathscr{P}$ are the sentence letters that appear across the top row of its tree.

MO The *main operator* of a formula $\mathscr{P}$ is the last operator added in its tree.

E2.3. For each of the following expressions, demonstrate that it is a formula and a sentence of $\mathcal{L}_s$ with a tree. Then on the tree (i) bracket all the subformulas, (ii) box the immediate subformula(s), (iii) star the atomic subformulas, and (iv) circle the main operator. A first case for $((\sim A \rightarrow B) \rightarrow A)$ is worked as an example.



These are formulas by FR(s)

From A, formula by FR($\sim$)

From $\sim A$ and B, formula by FR($\rightarrow$)

From $(\sim A \rightarrow B)$ and A, formula by FR($\rightarrow$)

*a. A

b. $\sim\sim\sim A$

c. $\sim(\sim A \rightarrow B)$

d. $(\sim C \rightarrow \sim(A \rightarrow \sim B))$

e. $(\sim(A \rightarrow B) \rightarrow (C \rightarrow \sim A))$

E2.4. Explain why the following expressions are not formulas or sentences of $\mathcal{L}_s$. Hint: you may find that an attempted tree will help you see what is wrong.

a. $(A \supset B)$

*b. $(\mathcal{P} \rightarrow \mathcal{Q})$

c. $(\sim B)$

d. $(A \rightarrow \sim B \rightarrow C)$

e. $((A \rightarrow B) \rightarrow \sim(A \rightarrow C) \rightarrow D)$

E2.5. For each of the following expressions, determine whether it is a formula and sentence of $\mathcal{L}_s$. If it is, show it on a tree, and exhibit its parts as in E2.3. If it is not, explain why as in E2.4.

*a. $\sim((A \rightarrow B) \rightarrow (\sim(A \rightarrow B) \rightarrow A))$

b. $\sim(A \rightarrow B \rightarrow (\sim(A \rightarrow B) \rightarrow A))$

*c. $\sim(A \rightarrow B) \rightarrow (\sim(A \rightarrow B) \rightarrow A)$

d. $\sim\sim\sim(\sim\sim\sim A \rightarrow \sim\sim\sim A)$

e. $((\sim(A \rightarrow B) \rightarrow (\sim C \rightarrow D)) \rightarrow \sim(\sim(E \rightarrow F) \rightarrow G))$

2.1.3 Abbreviations

We have completed the official grammar for our sentential languages. So far, the languages are relatively simple. For the purposes of later parts, when we turn to reasoning about logic, it will be good to have languages of this sort. However, for applications of logic, it will be advantageous to have additional expressions which, though redundant with expressions of the language already introduced, simplify the work. I begin by introducing these additional expressions, and then turn to the question about how to understand the redundancy.

Abbreviating. As may already be obvious, formulas of a sentential language like $\mathcal{L}_s$ can get complicated quickly. Abbreviated forms give us ways to manipulate official expressions without undue pain. First, for any formulas $\mathcal{P}$ and $\mathcal{Q}$,

AB (∨) $(\mathcal{P} \vee \mathcal{Q})$ abbreviates $(\sim\mathcal{P} \rightarrow \mathcal{Q})$

(∧) $(\mathcal{P} \wedge \mathcal{Q})$ abbreviates $\sim(\mathcal{P} \rightarrow \sim\mathcal{Q})$

(↔) $(\mathcal{P} \leftrightarrow \mathcal{Q})$ abbreviates $\sim((\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \rightarrow \mathcal{P}))$

The last of these is easier than it looks; I say something about this below. $\vee$ is *wedge*, $\wedge$ is *caret*, and $\leftrightarrow$ is *double arrow*. An expression of the form $(\mathcal{P} \vee \mathcal{Q})$ is a *disjunction* with $\mathcal{P}$ and $\mathcal{Q}$ as *disjuncts*; it has the standard reading, ($\mathcal{P}$ *or* $\mathcal{Q}$). An expression of the form $(\mathcal{P} \wedge \mathcal{Q})$ is a *conjunction* with $\mathcal{P}$ and $\mathcal{Q}$ as *conjuncts*; it has the standard reading, ($\mathcal{P}$ *and* $\mathcal{Q}$). An expression of the form $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *(material) biconditional*; it has the standard reading, ($\mathcal{P}$ *iff* $\mathcal{Q}$).[2] Again, we do not use ordinary English to define our symbols. All the same, this should suggest how the extra operators extend the range of what we are able to say in a natural way.

With the abbreviations, we are in a position to introduce derived clauses for FR. Suppose $\mathcal{P}$ and $\mathcal{Q}$ are formulas; then by FR(∼), $\sim\mathcal{P}$ is a formula; so by FR(→), $(\sim\mathcal{P} \rightarrow \mathcal{Q})$ is a formula; but this is just to say that $(\mathcal{P} \vee \mathcal{Q})$ is a formula. And similarly in the other cases. (If you are confused by such reasoning, work it out on a tree.) Thus we arrive at the following conditions.

FR′ (∨) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \vee \mathcal{Q})$ is a *formula*.

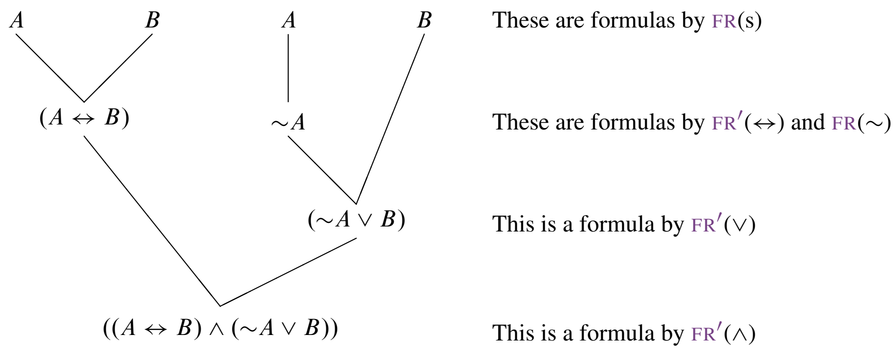
(∧) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \wedge \mathcal{Q})$ is a *formula*.

(↔) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *formula*.

[2] Common alternatives are & for $\wedge$, and $\equiv$ for $\leftrightarrow$.

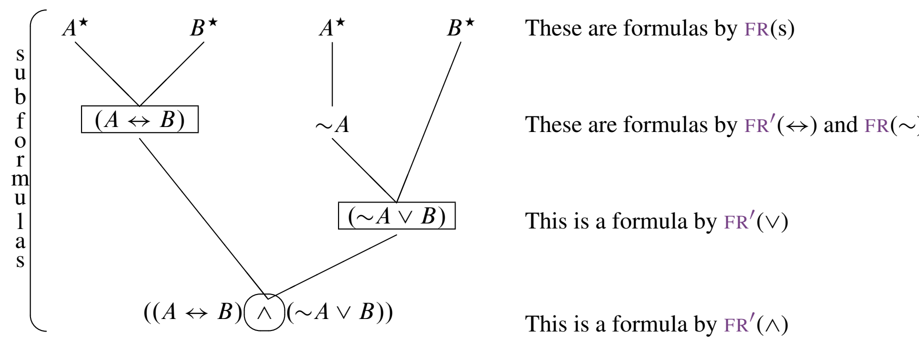
Once FR is extended in this way, the additional conditions may be applied directly in trees. Thus, for example, if $\mathcal{P}$ is a formula and $\mathcal{Q}$ is a formula, we can safely move in a tree to the conclusion that $(\mathcal{P} \vee \mathcal{Q})$ is a formula by FR$'(\vee)$. Similarly, for a more complex case, $((A \leftrightarrow B) \wedge (\sim A \vee B))$ is a formula.

(G)



In a derived sense, expressions with the new symbols have *subformulas*, *atomic* subformulas, *immediate* subformulas, and *main operator* all as before. Thus, with notation from exercises, with star for atomic formulas, box for immediate subformulas and circle for main operator, on the diagram immediately above,

(H)



In the derived sense, $((A \leftrightarrow B) \wedge (\sim A \vee B))$ has immediate subformulas $(A \leftrightarrow B)$ and $(\sim A \vee B)$, and main operator $\wedge$.

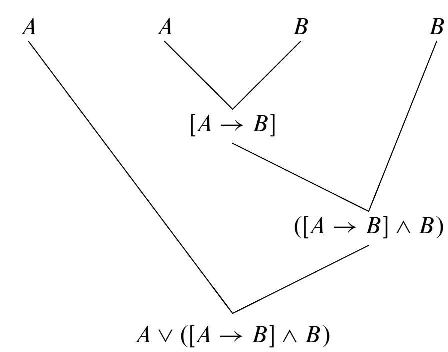
Return to the case of $(\mathcal{P} \leftrightarrow \mathcal{Q})$ and observe that it can be thought of as based on a simple abbreviation of the sort we expect. That is, $((\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P}))$ is of the sort $(\mathcal{A} \wedge \mathcal{B})$; so by AB$(\wedge)$, it abbreviates $\sim(\mathcal{A} \to \sim\mathcal{B})$; but with $(\mathcal{P} \to \mathcal{Q})$ for $\mathcal{A}$ and $(\mathcal{Q} \to \mathcal{P})$ for $\mathcal{B}$, this is just, $\sim((\mathcal{P} \to \mathcal{Q}) \to \sim(\mathcal{Q} \to \mathcal{P}))$ as in AB$(\leftrightarrow)$. So you may think of $(\mathcal{P} \leftrightarrow \mathcal{Q})$ as an abbreviation of $((\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P}))$, which in turn abbreviates the more complex $\sim((\mathcal{P} \to \mathcal{Q}) \to \sim(\mathcal{Q} \to \mathcal{P}))$.

A couple of additional abbreviations concern parentheses. First, it is sometimes convenient to use a pair of square brackets [ ] in place of parentheses ( ). This

is purely for visual convenience; for example ((()())) may be more difficult to absorb than ([()()]). Second, if the very last step of a tree for some formula $\mathcal{P}$ is justified by FR($\rightarrow$), FR$'$($\wedge$), FR$'$($\vee$), or FR$'$($\leftrightarrow$), we feel free to abbreviate $\mathcal{P}$ with the *outermost* set of parentheses or brackets dropped. Again, this is purely for visual convenience. Thus, for example, we might write, $A \rightarrow (B \rightarrow C)$ in place of $(A \rightarrow (B \rightarrow C))$. As it turns out, where $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are formulas, there is a difference between $((\mathcal{A} \rightarrow \mathcal{B}) \rightarrow \mathcal{C})$ and $(\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}))$, insofar as the main operator shifts from one case to the other. In $(\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C})$, however, it is not clear which arrow should be the main operator. That is why we do not count the latter as a grammatical formula or sentence. Similarly there is a difference between $\sim(\mathcal{A} \rightarrow \mathcal{B})$ and $(\sim\mathcal{A} \rightarrow \mathcal{B})$; again, the main operator shifts. However, there is no room for ambiguity when we drop just an outermost pair of parentheses and write $(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow \mathcal{C}$ for $((\mathcal{A} \rightarrow \mathcal{B}) \rightarrow \mathcal{C})$; and similarly when we write $\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})$ for $(\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}))$. And similarly for abbreviations with $\wedge$, $\vee$, or $\leftrightarrow$. So dropping outermost parentheses counts as a legitimate abbreviation.

An expression which uses the extra operators, square brackets, or drops outermost parentheses is a formula just insofar as it is a sort of shorthand for an official formula which does not. But we will not usually distinguish between the shorthand expressions and official formulas. Thus, again, the new conditions may be applied directly in trees and, for example, the following is a legitimate tree to demonstrate that $A \vee ([A \rightarrow B] \wedge B)$ is a formula.

(I)



Formulas by FR(s)

Formula by FR($\rightarrow$), with [ ]

Formula by FR$'$($\wedge$)

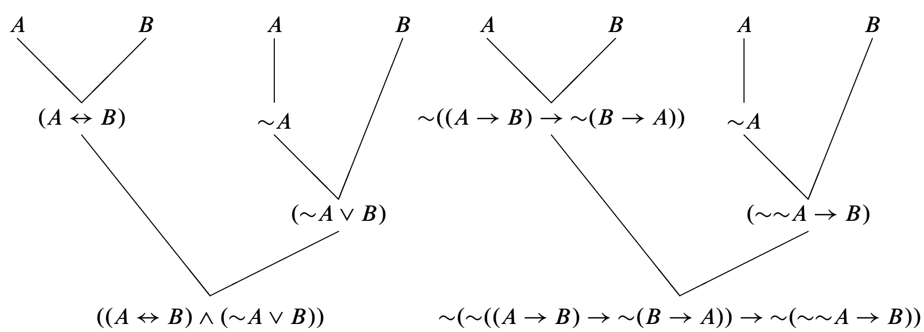
Formula by FR$'$($\vee$), with outer ( ) dropped

So we use our extra conditions for FR$'$, introduce square brackets instead of parentheses, and drop parentheses in the very last step. Remember that the *only* case where you can omit parentheses is if they would have been added in the very last step of the tree. So long as we do not distinguish between shorthand expressions and official formulas, we regard a tree of this sort as sufficient to demonstrate that an expression is a formula and a sentence.

Unabbreviating. As we have suggested, there is a certain tension between the advantages of a simple language, and one that is more complex. When a language is simple, it is easier to reason about; when it has additional resources, it is easier to use. Expressions with $\wedge$, $\vee$ and $\leftrightarrow$ are redundant with expressions that do not have them — though it is easier to work with a language that has $\wedge$, $\vee$ and $\leftrightarrow$ than with one that does not (something like reciting the Pledge of Allegiance in English, and then in Morse code; you can do it in either, but it is easier in the former). If all we wanted was a simple language to reason about, we would forget about the extra operators. If all we wanted was a language easy to use, we would forget about keeping the language simple. To have the advantages of both, we have adopted the position that expressions with the extra operators *abbreviate*, or are a shorthand for, expressions of the original language. It will be convenient to work with abbreviations in many contexts. But, when it comes to reasoning about the language, we set the abbreviations to the side, and focus on the official language itself.

For this to work, we have to be able to undo abbreviations when required. It is, of course, easy enough to substitute parentheses back for square brackets, or to replace outermost dropped parentheses. For formulas with the extra operators, it is always possible to work through trees, using AB to replace formulas with unabbreviated forms, one operator at a time. Consider an example.

(J)

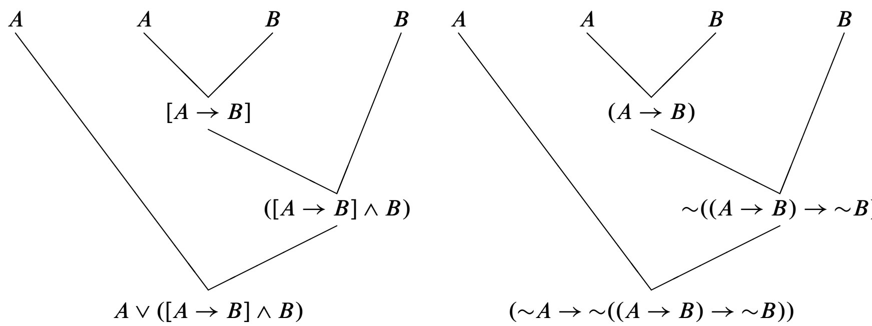


The tree on the left is (G) from above. The tree on the right simply includes "unpacked" versions of the expressions on the left. Atomics remain as before. Then, at each stage, given an unabbreviated version of the parts, we give an unabbreviated version of the whole. First, $(A \leftrightarrow B)$ abbreviates $\sim((A \to B) \to \sim(B \to A))$; this is a simple application of AB($\leftrightarrow$). $\sim A$ is not an abbreviation and so remains as before. From AB($\vee$), $(\mathcal{P} \vee \mathcal{Q})$ abbreviates $(\sim\mathcal{P} \to \mathcal{Q})$ so $(\sim A \vee B)$ abbreviates tilde the left disjunct, arrow the right (so that we get two tildes). For the final result, we combine the input formulas according to the unabbreviated form for $\wedge$. It is more a bookkeeping problem than anything: There is one formula $\mathcal{P}$ that is $(A \leftrightarrow B)$, another $\mathcal{Q}$ that is $(\sim A \vee B)$; these are combined into $(\mathcal{P} \wedge \mathcal{Q})$ and so, by AB($\wedge$), into

$\sim(\mathscr{P} \to \sim\mathscr{Q})$. You should be able to see that this is just what we have done. There is a tilde and a parenthesis; then the $\mathscr{P}$; then an arrow and a tilde; then the $\mathscr{Q}$, and a closing parenthesis. Not only is the abbreviation more compact but, as we shall see, there is a corresponding advantage when it comes to grasping what an expression says.

Here is a another example, this time from (I). In this case, we replace also square brackets and restore dropped outer parentheses.

(K)



In the right hand tree, we reintroduce parentheses for the square brackets. Similarly, we apply AB($\wedge$) and AB($\vee$) to unpack shorthand symbols. And outer parentheses are reintroduced at the very last step. Thus $([A \to B] \wedge B)$ is a shorthand for the unabbreviated expression, $(\sim A \to \sim((A \to B) \to \sim B))$.

Observe that right-hand trees are *not* ones of the sort you would use directly to show that an expression is a formula by FR! FR does not let you move directly from that $(A \to B)$ is a formula and B is a formula, to the result that $\sim((A \to B) \to \sim B)$ is a formula as just above. Of course, if $(A \to B)$ and B are formulas, then $\sim((A \to B) \to \sim B)$ is a formula, and nothing stops a tree to show it. This is the point of our derived clauses for FR$'$. In fact, this is a good check on your unabbreviations: If the result is not a formula, you have made a mistake! But you should not think of trees as on the right as involving application of FR. Rather they are *unabbreviating* trees, with application of AB to shorthand expressions from trees as on the left. A fully unabbreviated expression always meets all the requirements from section 2.1.2.

E2.6. For each of the following expressions, demonstrate that it is a formula and a sentence of $\mathscr{L}_s$ with a tree. Then on the tree (i) bracket all the subformulas, (ii) box the immediate subformula(s), (iii) star the atomic subformulas, and (iv) circle the main operator.

*a. $(A \wedge B) \to C$

b. $\sim([A \to \sim K_{14}] \vee C_3)$

c. $B \to (\sim A \leftrightarrow B)$

d. $(B \to A) \wedge (C \vee A)$

e. $(A \vee \sim B) \leftrightarrow (C \wedge A)$

*E2.7. For each of the formulas in E2.6a - e, produce an unabbreviating tree to find the unabbreviated expression it represents.

*E2.8. For each of the unabbreviated expressions from E2.7a - e, produce a complete tree to show by direct application of FR that it is an official formula.

E2.9. In the text, we introduced derived clauses to FR by reasoning as follows, "Suppose $\mathscr{P}$ and $\mathscr{Q}$ are formulas; then by FR($\sim$), $\sim\mathscr{P}$ is a formula; so by FR($\to$), $(\sim\mathscr{P} \to \mathscr{Q})$ is a formula; but this is just to say that $(\mathscr{P} \vee \mathscr{Q})$ is a formula. And similarly in the other cases" (p. 40). Supposing that $\mathscr{P}$ and $\mathscr{Q}$ are formulas, produce the similar reasoning to show that $(\mathscr{P} \wedge \mathscr{Q})$ and $(\mathscr{P} \leftrightarrow \mathscr{Q})$ are formulas. Hint: Again, it may help to think about trees.

E2.10. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The vocabulary for a sentential language, and use of the metalanguage.

b. A formula of a sentential language.

c. The parts of a formula.

d. The abbreviation and unabbreviation for an official formula of a sentential language.

2.2 Quantificational Languages

The *methods* by which we define the grammar of a quantificational language are very much the same as for a sentential language. Of course, in the quantificational case, additional expressive power is associated with additional complications. We will introduce a class of *terms* before we get to the formulas, and there will be a distinction between formulas and sentences — not all formulas are sentences. As before, however, we begin with the *vocabulary*; we then turn to the *terms*, *formulas*, and *sentences*. Again we conclude with some discussion of abbreviations.

Here is a brief intuitive picture. At the start of section 2.1 we introduced 'Bill is happy' and 'Hillary is happy' as atoms for sentential languages, and the rest of the section went on to fill out that picture. In this case, our atoms are certain sentence parts. Thus we introduce a class of *individual terms* which work to pick out objects. In the simplest case, these are like ordinary names such as 'Bill' and 'Hillary'; we will find it convenient to indicate these, b and h. Similarly, we introduce a class of *predicate* expressions as (x is happy) and (x loves y) indicating them by capitals as H^1 or L^2 (with the superscript to indicate the number of object *places*). Then H^1b says that Bill is happy, and L^2bh that Bill loves Hillary. We shall read $\forall xH^1x$ to say for any thing x *it* is happy — that *everything* is happy. (The upside-down 'A' for *all* is the *universal* quantifier.) As indicated by this reading, the variable x works very much like a pronoun in ordinary language. And, of course, our notions may be combined. Thus, $\forall xH^1x \wedge L^2hb$ says that everything is happy and Hillary loves Bill. Thus we expose structure buried in sentence letters from before. Of course we have so far done nothing to define quantificational languages. But this should give you a picture of the direction in which we aim to go.

2.2.1 Vocabulary

We begin by specifying the *vocabulary* or symbols of our quantificational languages. The vocabulary consists of infinitely many distinct symbols including,

VC (p) Punctuation symbols: ( )

(o) Operator symbols: $\sim$ $\rightarrow$ $\forall$

(v) Variable symbols: i j $\dots$ z with or without integer subscripts

(s) A possibly-empty countable collection of sentence letters

(c) A possibly-empty countable collection of constant symbols

(f) For any integer $n \geq 1$, a possibly-empty countable collection of n-place function symbols

(r) For any integer $n \geq 1$, a possibly-empty countable collection of n-place relation symbols

Unless otherwise noted, '$=$' is always included among the 2-place relation symbols. Notice that all the punctuation symbols, operator symbols and sentence letters remain from before (except that the collection of sentence letters may be empty). There is one new operator symbol, with the new variable symbols, constant symbols, function symbols, and relation symbols.

To fully specify the vocabulary of any particular language, we need to specify its sentence letters, constant symbols, function symbols, and relation symbols. Our general definition VC leaves room for languages with different collections of these symbols. As before, the requirement that the collections be countable is compatible with multiple series; for example, there may be sentence letters $A, A_1, A_2 \ldots, B, B_1, B_2 \ldots$ (where we may think of the unsubscripted letter as with an implicit subscript zero). So, again VC is compatible with a wide variety of specifications, but legitimate specifications always require that sentence letters, constant symbols, function symbols, and relation symbols can be sorted into series with a first member, a second member, and so forth. Notice that the *variable* symbols may be sorted into such a series as well.

$$\begin{array}{cccccccc} i & j & k & \ldots & z & i_1 & j_1 & \\ | & | & | & & | & | & | & \ldots \\ 0 & 1 & 2 & \ldots & 17 & 18 & 19 & \end{array}$$

So every variable is matched with an integer, and the variables are countable.

As a sample for the other symbols, we shall adopt a generic quantificational language $\mathcal{L}_q$ which includes the equality symbol '$=$' along with,

Sentence letters: uppercase Roman italics $A \ldots Z$ with or without integer subscripts

Constant symbols: lowercase Roman italics $a \ldots h$ with or without integer subscripts

Function symbols: for any integer $n \geq 1$, superscripted lowercase Roman italics $a^n \ldots z^n$ with or without integer subscripts

Relation symbols: for any integer $n \geq 1$, superscripted uppercase Roman italics $A^n \ldots Z^n$ with or without integer subscripts.

More on Countability

Given what was said on p. 33, one might think that every collection is countable. However, this is not so. This amazing and simple result was proved by G. Cantor in 1873. Consider the collection which includes every countably infinite series of digits 0 through 9 (or, if you like, the collection of all real numbers between 0 and 1). Suppose that the members of this collection can be correlated one-to-one with the integers. Then there is some list,

0	–	$\boldsymbol{a_0}$	a_1	a_2	a_3	a_4	…
1	–	b_0	$\boldsymbol{b_1}$	b_2	b_3	b_4	…
2	–	c_0	c_1	$\boldsymbol{c_2}$	c_3	c_4	…
3	–	d_0	d_1	d_2	$\boldsymbol{d_3}$	d_4	…
4	–	e_0	e_1	e_2	e_3	$\boldsymbol{e_4}$	…

and so forth, which matches each series of digits with an integer. For any digit x, say x' is the digit after it in the standard ordering (where 0 follows 9). Now consider the digits along the diagonal, $a_0, b_1, c_2, d_3, e_4 \ldots$ and ask: does the series $a'_0, b'_1, c'_2, d'_3, e'_4 \ldots$ appear anywhere in the list? It cannot be the first member, because $a_0 \neq a'_0$; it cannot be the second, because $b_1 \neq b'_1$, and similarly for every member! So $a'_1, b'_2, c'_3, d'_4, e'_5 \ldots$ does not appear in the list. So we have *failed* to match *all* the infinite series of digits with integers — and similarly for *any* attempt! So the collection which contains every countably infinite series of digits is not countable.

As an example, consider the following attempt to line up the integers with the series of digits:

0	–	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	…
1	–	1	**1**	1	1	1	1	1	1	1	1	1	1	1	1	…
2	–	2	2	**2**	2	2	2	2	2	2	2	2	2	2	2	…
3	–	3	3	3	**3**	3	3	3	3	3	3	3	3	3	3	…
4	–	4	4	4	4	**4**	4	4	4	4	4	4	4	4	4	…
5	–	5	5	5	5	5	**5**	5	5	5	5	5	5	5	5	…
6	–	6	6	6	6	6	6	**6**	6	6	6	6	6	6	6	…
7	–	7	7	7	7	7	7	7	**7**	7	7	7	7	7	7	…
8	–	8	8	8	8	8	8	8	8	**8**	8	8	8	8	8	…
9	–	9	9	9	9	9	9	9	9	9	**9**	9	9	9	9	…
10	–	1	0	1	0	1	0	1	0	1	0	**1**	0	1	0	…
11	–	0	1	1	1	1	1	1	1	1	1	1	**1**	1	1	…
12	–	1	2	1	2	1	2	1	2	1	2	1	2	**1**	2	…
13	–	1	3	1	3	1	3	1	3	1	3	1	3	1	**3**	…

and so forth. For each integer, repeat its digits, except that for "duplicate" cases — 1 and 11, 2 and 22, 12 and 1212 — prefix enough 0s so that no later series duplicates an earlier one. Then, by the above method, from the diagonal,

1 2 3 4 5 6 7 8 9 0 2 2 2 4 …

cannot appear anywhere on the list. And similarly, any list has some missing series.

Observe that constant symbols and variable symbols partition the lowercase alphabet: $a \dots h$ for constants, and $i \dots z$ for variables. Sentence letters are distinguished from relation symbols by superscripts; similarly, constant and variable symbols are distinguished from function symbols by superscripts. Function symbols with a superscript 1 ($a^1 \dots z^1$) are *one-place* function symbols; function symbols with a superscript 2 ($a^2 \dots z^2$) are *two-place* function symbols; and so forth. Similarly, relation symbols with a superscript 1 ($A^1 \dots Z^1$) are *one-place* relation symbols; relation symbols with a superscript 2 ($A^2 \dots Z^2$) are *two-place* relation symbols; and so forth. Subscripts merely guarantee that we never run out of symbols of the different types. Notice that superscripts and subscripts suffice to distinguish all the different symbols from one another. Thus, for example A and A^1 are different symbols — one a sentence letter, and the other a one-place relation symbol; A^1, A^1_1 and A^2 are distinct as well — the first two are one-place relation symbols, distinguished by the subscript; the latter is a completely distinct two-place relation symbol. In practice, again, we will not see subscripts very often. (And we shall even find ways to abbreviate away some superscripts.)

The *metalanguage* works very much as before. We use script letters $\mathcal{A} \dots \mathcal{Z}$ and $\mathscr{a} \dots \mathscr{z}$ to represent expressions of an object language like $\mathcal{L}_q$. Again, '$\sim$', '$\to$', '$\forall$', '$=$', '(', and ')' represent themselves. And concatenated or joined symbols of the metalanguage represent the concatenation of the symbols they represent. As before, the metalanguage lets us make general claims about ranges of expressions all at once. Thus, where $\mathscr{x}$ is a variable, $\forall \mathscr{x}$ is a *universal* $\mathscr{x}$*-quantifier*. Here, $\forall \mathscr{x}$ is not an expression of an object language like $\mathcal{L}_q$ (Why?) Rather, we have said of object language expressions that $\forall \mathscr{x}$ is a universal $\mathscr{x}$-quantifier, $\forall y_2$ is a universal y_2-quantifier, and so forth. In the metalinguistic expression, '$\forall$' stands for itself, and '$\mathscr{x}$' for the arbitrary variable. Again, as in section 2.1.1, it may help to use maps to see whether an expression is of a given form. Thus given that $\mathscr{x}$ maps to any variable, $\forall x$ and $\forall y$ are of the form $\forall \mathscr{x}$, but $\forall c$ and $\forall f^1 z$ are not.

$$\text{(L)} \qquad \begin{array}{cccc} \forall \mathscr{x} & \forall \mathscr{x} & \forall \mathscr{x} & \forall \mathscr{x} \\ \downarrow\ \downarrow & \downarrow\ \downarrow & \downarrow & \downarrow\quad \overbrace{}^{?} \\ \forall x & \forall y & \forall c^{?} & \forall\ \ f^1 z \end{array}$$

In the leftmost two cases, $\forall$ maps to itself, and $\mathscr{x}$ to a variable. In the next, 'c' is a *constant* so there is no variable to which $\mathscr{x}$ can map. In the rightmost case, there is a variable z in the object expression, but if $\mathscr{x}$ is mapped to it, the function symbol f^1 is left unmatched. So the rightmost two expressions are not of the form $\forall \mathscr{x}$.

E2.11. Assuming that $\mathcal{R}^1$ may represent any one-place relation symbol, $\mathscr{h}^2$ any two-place function symbol, $\mathscr{x}$ any variable, and $\mathscr{c}$ any constant of $\mathcal{L}_q$, use maps to determine whether each of the following expressions is (i) of the form, $\forall \mathscr{x}(\mathcal{R}^1\mathscr{x} \to \mathcal{R}^1\mathscr{c})$ and then (ii) of the form, $\forall \mathscr{x}(\mathcal{R}^1\mathscr{x} \to \mathcal{R}^1\mathscr{h}^2\mathscr{x}\mathscr{c})$.

a. $\forall k(A^1k \to A^1d)$

b. $\forall h(J^1h \to J^1b)$

c. $\forall w(S^1w \to S^1g^2wb)$

d. $\forall w(S^1w \to S^1c^2xc)$

e. $\forall vL^1v \to L^1yh^2$

2.2.2 Terms

With the vocabulary of a language in place, we can turn to specification of its grammatical expressions. For this, in the quantificational case, we begin with *terms*.

TR (v) If $\mathscr{t}$ is a variable $\mathscr{x}$, then $\mathscr{t}$ is a *term*.

(c) If $\mathscr{t}$ is a constant $\mathscr{c}$, then $\mathscr{t}$ is a *term*.

(f) If $\mathscr{h}^n$ is a n-place function symbol and $\mathscr{t}_1 \dots \mathscr{t}_n$ are n terms, then $\mathscr{h}^n\mathscr{t}_1 \dots \mathscr{t}_n$ is a *term*.

(CL) Any term may be formed by repeated application of these rules.

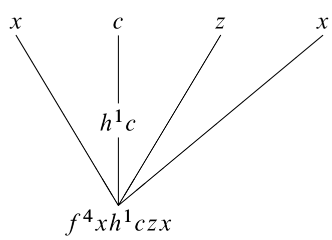
TR is another example of a recursive definition. As before, we can use tree diagrams to see how it works. This time, basic elements are constants and variables. Complex elements are put together by clause (f). Thus, for example, $f^1g^2h^1xc$ is a term of $\mathcal{L}_q$.

(M)

x $\quad$ c	x is a term by TR(v), and c is a term by TR(c)
h^1x	since x is a term, this is a term by TR(f)
g^2h^1xc	since h^1x and c are terms, this is a term by TR(f)
$f^1g^2h^1xc$	since g^2h^1xc is a term, this is a term by TR(f)

Notice how the superscripts of a function symbol indicate the number of places that take terms. Thus x is a term, and h^1 followed by x to form h^1x is another term. But then, given that h^1x and c are terms, g^2 followed by h^1x and then c is another term. And so forth. Observe that neither g^2h^1x nor g^2c are terms — the function symbol g^2 must be followed by a pair of terms to form a new term. And neither is h^1xc a term — the function symbol h^1 can only be followed by a single term to compose a term. You will find that there is always only one way to build a term on a tree. Here is another example.

(N)



these are terms by TR(v), TR(c), TR(v), and TR(v)

since c is a term, this is a term by TR(f)

given the four input terms, this is a term by TR(f)

Again, there is always just one way to build a term by the definition. If you are confused about the makeup of a term, build it on a tree, and all will be revealed. To demonstrate that an expression is a term, it is sufficient to construct it, according to the definition, on such a tree. If an expression is not a term, there will be no way to construct it according to the rules.

E2.12. For each of the following expressions, demonstrate that it is a term of $\mathscr{L}_q$ with a tree.

a. f^1c

b. g^2yf^1c

*c. h^3cf^1yx

d. $g^2h^3xyf^1cx$

e. $h^3f^1f^1xcg^2f^1za$

E2.13. Explain why the following expressions are not terms of $\mathscr{L}_q$. Hint: you may find that an attempted tree will help you see what is wrong.

a. X

b. g^2

c. zc

*d. $g^2 y f^1 xc$

e. $h^3 f^1 f^1 cg^2 f^1 za$

E2.14. For each of the following expressions, determine whether it is a term of $\mathcal{L}_q$; if it is, demonstrate with a tree; if not, explain why.

*a. $g^2 g^2 xyf^1 x$

*b. $h^3 cf^2 yx$

c. $f^1 g^2 xh^3 yf^2 yc$

d. $f^1 g^2 xh^3 yf^1 yc$

e. $h^3 g^2 f^1 xcg^2 f^1 zaf^1 b$

2.2.3 Formulas

With the terms in place, we are ready for the central notion of a formula. Again, the definition is recursive.

FR (s) If $\mathcal{S}$ is a sentence letter, then $\mathcal{S}$ is a *formula*.

(r) If $\mathcal{R}^n$ is an n-place relation symbol and $t_1 \dots t_n$ are n terms, then $\mathcal{R}^n t_1 \dots t_n$ is a *formula*.

($\sim$) If $\mathcal{P}$ is a formula, then $\sim\mathcal{P}$ is a *formula*.

($\rightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \rightarrow \mathcal{Q})$ is a *formula*.

($\forall$) If $\mathcal{P}$ is a formula and x is a variable, then $\forall x\mathcal{P}$ is a *formula*.

(CL) Any formula can be formed by repeated application of these rules.

Again, we can use trees to see how it works. In this case, FR(r) depends on which expressions are terms. So it is natural to split the diagram into two, with applications of TR above a division, and FR below. Then, for example, $\forall x(A^1 f^1 x \rightarrow \sim\forall y B^2 cy)$ is a formula.

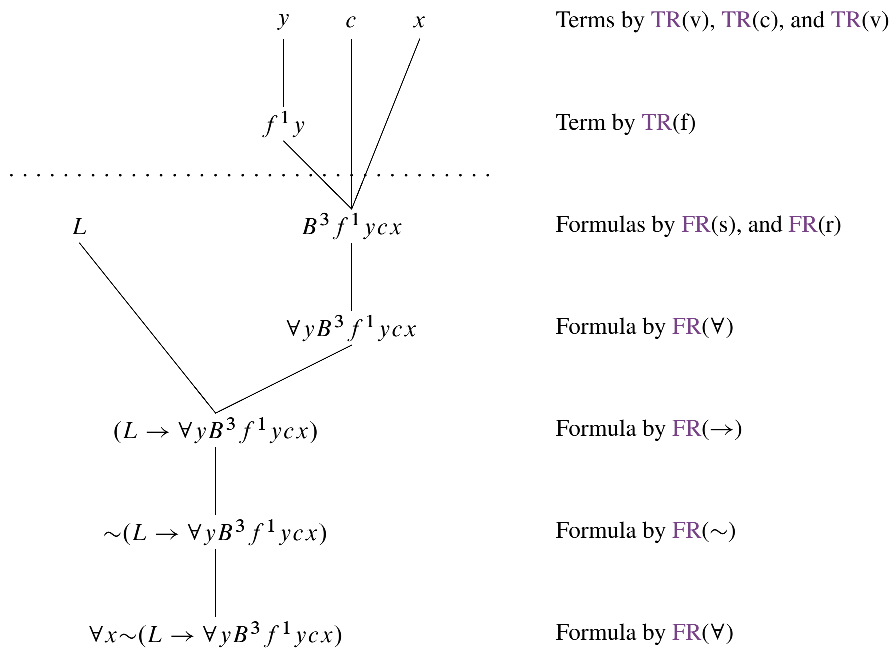
(O)

$x \quad c \quad y$	Terms by TR(v), TR(c), and TR(v)
f^1x	Term by TR(f)
$A^1f^1x \quad B^2cy$	Formulas by FR(r)
$\forall yB^2cy$	Formula by FR(∀)
$\sim\forall yB^2cy$	Formula by FR(∼)
$(A^1f^1x \to \sim\forall yB^2cy)$	Formula by FR(→)
$\forall x(A^1f^1x \to \sim\forall yB^2cy)$	Formula by FR(∀)

By now, the basic strategy should be clear. We construct terms by TR just as before. Given that f^1x is a term, FR(r) gives us that A^1f^1x is a formula, for it consists of a one-place relation symbol followed by a single term; and given that c and y are terms, FR(r) gives us that B^2cy is a formula, for it consists of a two-place relation symbol followed by two terms. From the latter, by FR(∀), $\forall yB^2cy$ is a formula. Then FR(∼) and FR(→) work just as before. The final step is another application of FR(∀).

Here is another example. By the following tree, $\forall x{\sim}(L \to \forall yB^3f^1ycx)$ is a formula of $\mathscr{L}_q$.

(P)



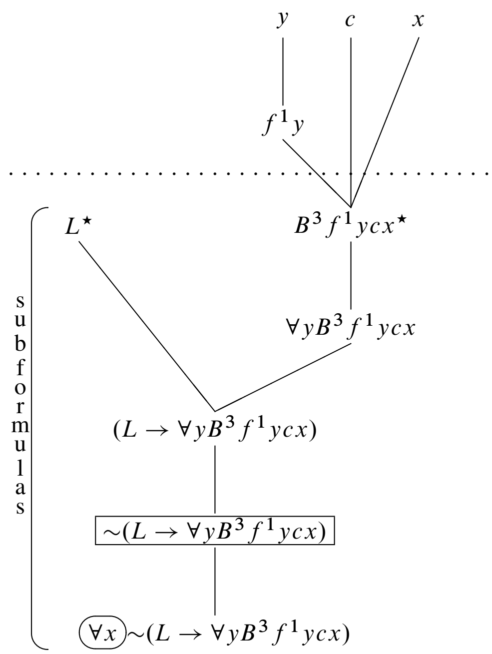
The basic formulas appear in the top row of the formula part of the diagram. L is a sentence letter. So it does not require any terms to be a formula. B^3 is a three-place relation symbol, so by FR(r) it takes three terms to make a formula. After that, other formulas are constructed out of ones that come before.

If an expression is not a formula, then there is no way to construct it by the rules. Thus, for example, (A^1x) is not a formula of $\mathcal{L}_q$. A^1x is a formula; but the only way parentheses are introduced is in association with →; the parentheses in (A^1x) are not introduced that way; so there is no way to construct it by the rules, and it is not a formula. Similarly, A^2x and A^2f^2xy are not formulas; in each case, the problem is that the two-place relation symbol is followed by just *one* term. You should be clear about these in your own mind, particularly for the second case.

Before turning to the official notion of a *sentence*, we introduce some additional definitions, each directly related to the trees — and to notions you have seen before. First, where '→', '∼', and any quantifier count as *operators*, a formula's *main* operator is the last operator added in its tree. Second, every formula in the formula portion of a diagram for $\mathcal{P}$, including $\mathcal{P}$ itself, is a *subformula* of $\mathcal{P}$. Notice that terms are not formulas, and so are not subformulas. An *immediate* subformula of $\mathcal{P}$ is a subformula to which $\mathcal{P}$ is directly connected by lines. A subformula is *atomic* iff it contains no operators and so appears in the top line of the formula part of the tree. Thus, with notation from exercises before, with star for atomic formulas, box

for immediate subformulas and circle for main operator, on the diagram immediately above we have,

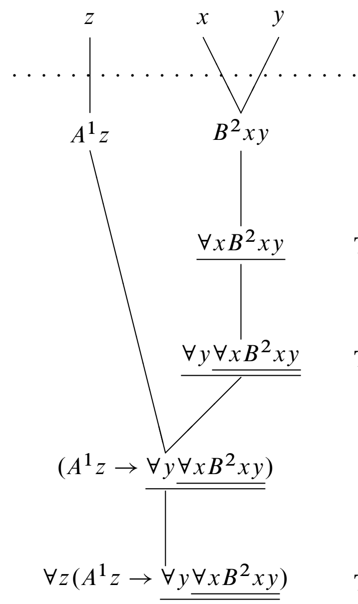
(Q)



The main operator is $\forall x$, and the immediate subformula is $\sim(L \to \forall yB^3f^1ycx)$. The atomic subformulas are L and B^3f^1ycx. The atomic subformulas are the most basic *formulas*. Given this, everything is as one would expect from before. In general, if $\mathcal{P}$ and $\mathcal{Q}$ are formulas and $\mathscr{x}$ is a variable, the main operator of $\forall \mathscr{x}\mathcal{P}$ is the quantifier, and the immediate subformula is $\mathcal{P}$; the main operator of $\sim\mathcal{P}$ is the tilde, and the immediate subformula is $\mathcal{P}$; the main operator of $(\mathcal{P} \to \mathcal{Q})$ is the arrow, and the immediate subformulas are $\mathcal{P}$ and $\mathcal{Q}$ — for you would build these by getting $\mathcal{P}$, or $\mathcal{P}$ and $\mathcal{Q}$, and then adding the quantifier, tilde, or arrow as the last operator.

Now if a formula includes a quantifier, that quantifier's *scope* is just the subformula in which the quantifier *first* appears. Using underlines to indicate scope,

(R)

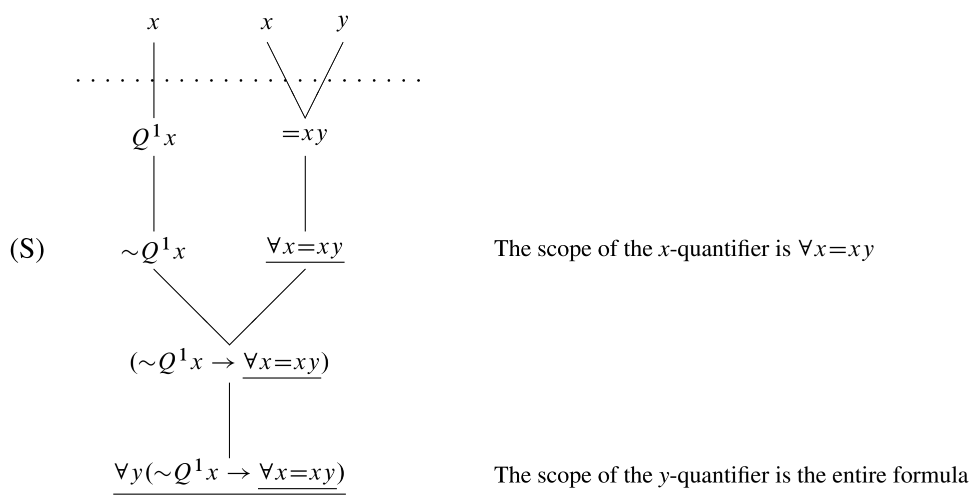


The scope of the x-quantifier is $\forall xB^2xy$

The scope of the y-quantifier is $\forall y\forall xB^2xy$

The scope of the z-quantifier is the entire formula

A variable x is *bound* iff it appears in the scope of an x quantifier, and a variable is *free* iff it is not bound. In the above diagram, each variable is bound. The x-quantifier binds both instances of x; the y-quantifier binds both instances of y; and the z-quantifier binds both instances of z. In $\forall xR^2xy$, however, both instances of x are bound, but the y is free. Finally, and expression is a *sentence* iff it is a formula and it has no free variables. To determine whether an expression is a sentence, use a tree to see if it is a formula. If it is a formula, use underlines to check whether any variable x has an instance that falls outside the scope of an x-quantifier. If it is a formula, and there is no such instance, then the expression is a sentence. From the above diagram, $\forall z(A^1z \rightarrow \forall y\forall xB^2xy)$ is a formula and a sentence. But as follows, $\forall y(\sim Q^1x \rightarrow \forall x{=}xy)$ is not.



Recall that '$=$' is a two-place relation symbol. The expression has a tree, so it is a formula. The *x*-quantifier binds the last two instances of *x*, and the *y*-quantifier binds both instances of y. But the first instance of x is free. Since it has a free variable, although it is a formula, $\forall y(\sim Q^1x \rightarrow \forall x{=}xy)$ is not a sentence. Notice that $\forall xR^2ax$, for example, is a sentence, as the only *variable* is x (a being a *constant*) and all the instances of x are bound.

E2.15. For each of the following expressions, (i) Demonstrate that it is a formula of $\mathscr{L}_q$ with a tree. (ii) On the tree bracket all the subformulas, box the immediate subformulas, star the atomic subformulas, circle the main operator, and indicate quantifier scope with underlines. Then (iii) say whether the formula is a sentence, and if it is not, explain why.

a. H^1x

*b. B^2ac

c. $\forall x(\sim{=}xc \rightarrow A^1g^2ay)$

d. $\sim\forall x(B^2xc \rightarrow \forall y\sim A^1g^2ay)$

e. $(S \rightarrow \sim(\forall wB^2f^1wh^1a \rightarrow \sim\forall z(H^1w \rightarrow B^2za)))$

E2.16. Explain why the following expressions are not formulas or sentences of $\mathscr{L}_q$. Hint: You may find that an attempted tree will help you see what is wrong.

a. H^1

b. g^2ax

*c. $\forall x B^2 x g^2 a x$

d. $\sim(\sim\forall a A^1 a \to (S \to \sim B^2 z g^2 x a))$

e. $\forall x (Dax \to \forall z \sim K^2 z g^2 x a)$

E2.17. For each of the following expressions, determine whether it is a formula and a sentence of $\mathcal{L}_q$. If it is a formula, show it on a tree, and exhibit its parts as in E2.15. If it fails one or both, explain why.

a. $\sim(L \to \sim V)$

b. $\forall x(\sim L \to K^1 h^3 x b)$

c. $\forall z \forall w(\forall x R^2 w x \to \sim K^2 z w) \to \sim M^2 z z)$

*d. $\forall z(L^1 z \to (\forall w R^2 w f^3 a x w \to \forall w R^2 f^3 a z w w))$

e. $\sim((\forall w) B^2 f^1 w h^1 a \to \sim(\forall z)(H^1 w \to B^2 z a))$

2.2.4 Abbreviations

That is all there is to the official grammar. Having introduced the official grammar, though, it is nice to have in hand some abbreviated versions for official expressions. Abbreviated forms give us ways to manipulate official expressions without undue pain. First, for any variable x and formulas $\mathcal{P}$ and $\mathcal{Q}$,

AB ($\vee$) $(\mathcal{P} \vee \mathcal{Q})$ abbreviates $(\sim\mathcal{P} \to \mathcal{Q})$

($\wedge$) $(\mathcal{P} \wedge \mathcal{Q})$ abbreviates $\sim(\mathcal{P} \to \sim\mathcal{Q})$

($\leftrightarrow$) $(\mathcal{P} \leftrightarrow \mathcal{Q})$ abbreviates $\sim((\mathcal{P} \to \mathcal{Q}) \to \sim(\mathcal{Q} \to \mathcal{P}))$

($\exists$) $\exists x\mathcal{P}$ abbreviates $\sim\forall x\sim\mathcal{P}$

The first three are as from AB. The last is new. For any variable x, an expression of the form $\exists x$ is an *existential* quantifier — it is read to say, there *exists* an x such that $\mathcal{P}$.

As before, these abbreviations make possible derived clauses to FR. Suppose $\mathcal{P}$ is a formula; then by FR($\sim$), $\sim\mathcal{P}$ is a formula; so by FR($\forall$), $\forall x\sim\mathcal{P}$ is a formula; so by FR($\sim$) again, $\sim\forall x\sim\mathcal{P}$ is a formula; but this is just to say that $\exists x\mathcal{P}$ is a formula. With results from before, we are thus given,

FR′ (∧) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \wedge \mathcal{Q})$ is a *formula*.

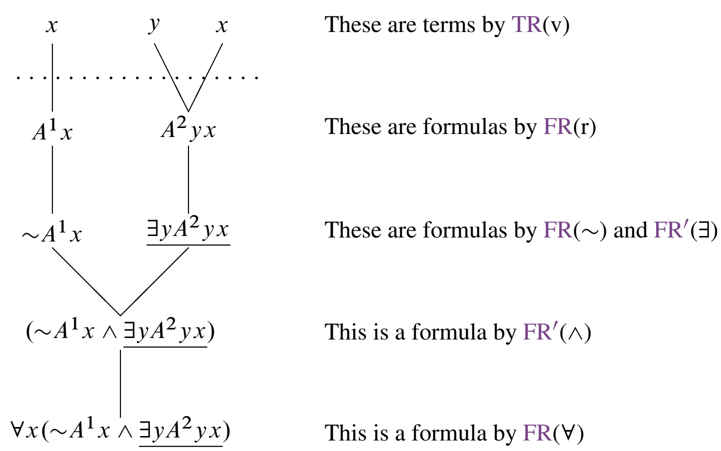
(∨) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \vee \mathcal{Q})$ is a *formula*.

(↔) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *formula*.

(∃) If $\mathcal{P}$ is a formula and x is a variable, then $\exists x\mathcal{P}$ is a *formula*.

The first three are from before. The last is new. And, as before, we can incorporate these conditions directly into trees for formulas. Thus $\forall x(\sim A^1x \wedge \exists yA^2yx)$ is a formula.

(T)



In a derived sense, we carry over additional definitions from before. Thus, the main operator is the last operator added in its tree, subformulas are all the formulas in the formula part of a tree, atomic subformulas are the ones in the upper row of the formula part, and immediate subformulas are the one(s) to which a formula is directly connected by lines. Thus the main operator of $\forall x(\sim A^1x \wedge \exists yA^2yx)$ is the universal quantifier and the immediate subformula is $(\sim A^1x \wedge \exists yA^2yx)$. In addition, a variable is in the scope of an existential quantifier iff it would be in the scope of the unabbreviated universal one. So it is possible to discover whether an expression is a sentence directly from diagrams of this sort. Thus, as indicated by underlines, $\forall x(\sim A^1x \wedge \exists yA^2yx)$ is a sentence.

To see what it is an abbreviation for, we can reconstruct the formula on an unabbreviating tree, one operator at a time.

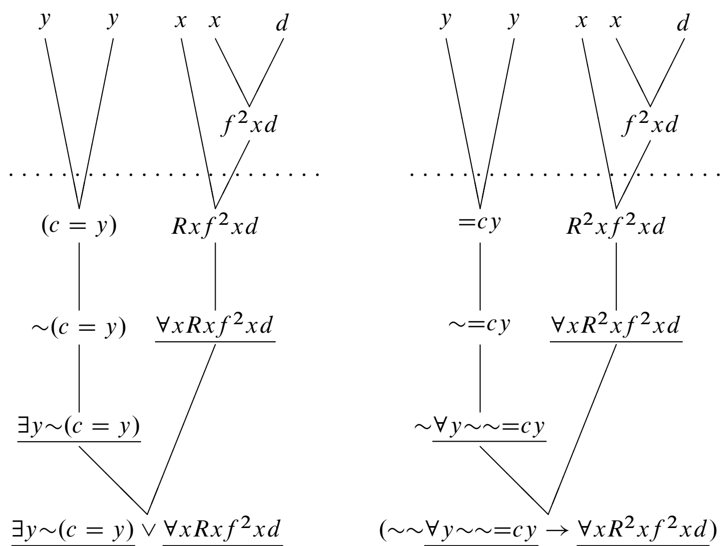
$$
\begin{array}{lcccc}
 & x \qquad y \quad x & & x \qquad y \quad x & \\
 & A^1x \qquad A^2yx & & A^1x \qquad A^2yx & \\
\text{(U)} & {\sim}A^1x \qquad \underline{\exists yA^2yx} & & {\sim}A^1x \qquad {\sim}\underline{\forall y{\sim}A^2yx} & \text{By AB}(\exists) \\
 & ({\sim}A^1x \wedge \underline{\exists yA^2yx}) & & {\sim}({\sim}A^1x \rightarrow {\sim}{\sim}\underline{\forall y{\sim}A^2yx}) & \text{By AB}(\wedge) \\
 & \underline{\forall x({\sim}A^1x \wedge \underline{\exists yA^2yx})} & & \underline{\forall x{\sim}({\sim}A^1x \rightarrow {\sim}{\sim}\underline{\forall y{\sim}A^2yx})} &
\end{array}
$$

First the existential quantifier is replaced by the unabbreviated form. Then, where $\mathcal{P}$ and $\mathcal{Q}$ are joined by FR$'(\wedge)$ to form $(\mathcal{P} \wedge \mathcal{Q})$, the corresponding unabbreviated expressions are combined into the unabbreviated form, ${\sim}(\mathcal{P} \rightarrow {\sim}\mathcal{Q})$. At the last step, FR$(\forall)$ applies as before. So $\forall x({\sim}A^1x \wedge \exists yA^2yx)$ abbreviates $\forall x{\sim}({\sim}A^1x \rightarrow {\sim}{\sim}\forall y{\sim}A^2yx)$. Again, abbreviations are nice! Notice that the resultant expression is a formula and a sentence, as it should be.

As before, it is sometimes convenient to use a pair of square brackets [ ] in place of parentheses ( ). And if the very last step of a tree for some formula is justified by FR$(\rightarrow)$, FR$'(\vee)$, FR$'(\wedge)$, or FR$'(\leftrightarrow)$, we may abbreviate that formula with the outermost set of parentheses or brackets dropped. In addition, for terms t_1 and t_2 we will frequently represent the formula $=t_1t_2$ as $(t_1 = t_2)$. Notice the extra parentheses. This lets us see the equality symbol in its more usual "infix" form. When there is no danger of confusion, we will sometimes omit the parentheses and write, $t_1 = t_2$. Also, where there is no potential for confusion, we sometimes omit superscripts. Thus in $\mathcal{L}_q$ we might omit superscripts on relation symbols — simply assuming that the terms following a relation symbol give its correct number of places. Thus Ax abbreviates A^1x; Axy abbreviates A^2xy; Axf^1y abbreviates A^2xf^1y; and so forth. Notice that Ax and Axy, for example, involve *different* relation symbols. In formulas of $\mathcal{L}_q$, sentence letters are distinguished from relation symbols insofar as relation symbols are followed immediately by terms, where sentence letters are not. Notice, however, that we *cannot* drop superscripts on function symbols in $\mathcal{L}_q$ — thus, even given that f and g are function symbols rather than constants, apart from superscripts, there is no way to distinguish the terms in, say, $Afgxyzw$.

As a final example, $\exists y{\sim}(c = y) \vee \forall xRxf^2xd$ is a formula and a sentence.

(V)



The abbreviation drops a superscript, uses the infix notation for equality, uses the existential quantifier and wedge, and drops outermost parentheses. As before, the right-hand diagram is not a direct demonstration that $(\sim\sim\forall y\sim\sim=cy \to \forall xR^2xf^2xd)$ is a sentence. However, it unpacks the abbreviation, and we know that the result is an official sentence, insofar as the left-hand tree, with its application of derived rules, tells us that $\exists y\sim(c = y) \vee \forall xRxf^2xd$ is an abbreviation of formula and a sentence, and the right-hand diagram tells us what that expression is.

E2.18. For each of the following expressions, (i) Demonstrate that it is a formula of $\mathcal{L}_q$ with a tree. (ii) On the tree bracket all the subformulas, box the immediate subformulas, star the atomic subformulas, circle the main operator, and indicate quantifier scope with underlines. Then (iii) say whether the formula is a sentence, and if it is not, explain why.

a. $(A \to \sim B) \leftrightarrow (A \wedge C)$

b. $\exists xFx \wedge \forall yGxy$

*c. $\exists xAf^1g^2ah^3zwf^1x \vee S$

d. $\forall x\forall y\forall z([(x = y) \wedge (y = z)] \to (x = z))$

e. $\exists y[c = y \wedge \forall xRxf^1xy]$

Grammar Quick Reference

VC
- (p) Punctuation symbols: (,)
- (o) Operator symbols: $\sim$, $\rightarrow$, $\forall$
- (v) Variable symbols: $i \ldots z$ with or without integer subscripts
- (s) A possibly-empty countable collection of sentence letters
- (c) A possibly-empty countable collection of constant symbols
- (f) For any integer $n \geq 1$, a possibly-empty countable collection of n-place function symbols
- (r) For any integer $n \geq 1$, a possibly-empty countable collection of n-place relation symbols

TR
- (v) If t is a variable x, then t is a *term*.
- (c) If t is a constant c, then t is a *term*.
- (f) If h^n is a n-place function symbol and $t_1 \ldots t_n$ are n terms, then $h^n t_1 \ldots t_n$ is a *term*.
- (CL) Any term may be formed by repeated application of these rules.

FR
- (s) If $\mathcal{S}$ is a sentence letter, then $\mathcal{S}$ is a *formula*.
- (r) If $\mathcal{R}^n$ is an n-place relation symbol and $t_1 \ldots t_n$ are n terms, $\mathcal{R}^n t_1 \ldots t_n$ is a *formula*.
- ($\sim$) If $\mathcal{P}$ is a formula, then $\sim\mathcal{P}$ is a *formula*.
- ($\rightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \rightarrow \mathcal{Q})$ is a *formula*.
- ($\forall$) If $\mathcal{P}$ is a formula and x is a variable, then $\forall x\mathcal{P}$ is a *formula*.
- (CL) Any formula can be formed by repeated application of these rules.

A quantifier's *scope* includes just the formula on which it is introduced; a variable x is *free* iff it is not in the scope of an x-quantifier; an expression is a *sentence* iff it is a formula with no free variables. A formula's *main operator* is the last operator added; its *immediate* subformulas are the ones to which it is directly connected by lines.

AB
- ($\vee$) $(\mathcal{P} \vee \mathcal{Q})$ abbreviates $(\sim\mathcal{P} \rightarrow \mathcal{Q})$
- ($\wedge$) $(\mathcal{P} \wedge \mathcal{Q})$ abbreviates $\sim(\mathcal{P} \rightarrow \sim\mathcal{Q})$
- ($\leftrightarrow$) $(\mathcal{P} \leftrightarrow \mathcal{Q})$ abbreviates $\sim((\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \rightarrow \mathcal{P}))$
- ($\exists$) $\exists x\mathcal{P}$ abbreviates $\sim\forall x\sim\mathcal{P}$

FR′
- ($\wedge$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \wedge \mathcal{Q})$ is a *formula*.
- ($\vee$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \vee \mathcal{Q})$ is a *formula*.
- ($\leftrightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $(\mathcal{P} \leftrightarrow \mathcal{Q})$ is a *formula*.
- ($\exists$) If $\mathcal{P}$ is a formula and x is a variable, then $\exists x\mathcal{P}$ is a *formula*.

The generic language $\mathcal{L}_q$ includes the equality symbol '=' along with,

Sentence letters: $A \ldots Z$ with or without integer subscripts

Constant symbols: $a \ldots h$ with or without integer subscripts

Function symbols: for any $n \geq 1$, $a^n \ldots z^n$ with or without integer subscripts

Relation symbols: for any $n \geq 1$, $A^n \ldots Z^n$ with or without integer subscripts.

*E2.19. For each of the formulas in E2.18, produce an unabbreviating tree to find the unabbreviated expression it represents.

*E2.20. For each of the unabbreviated expressions from E2.19, produce a complete tree to show by direct application of FR that it is an official formula. In each case, using underlines to indicate quantifier scope, is the expression a sentence? does this match with the result of E2.18?

2.2.5 Another Language

To emphasize the generality of our definitions VC, TR, and FR, let us introduce an enhanced version of a language with which we will be much concerned later in the text. $\mathscr{L}_{NT}^{<}$ is like a minimal language we shall introduce later for *number theory*. Recall that VC leaves open what are the sentence letters, constant symbols, function symbols and relation symbols of a quantificational language. So far, our generic language $\mathscr{L}_q$ fills these in by certain conventions. $\mathscr{L}_{NT}^{<}$ replaces these with,

Constant symbol: $\emptyset$

two-place relation symbols: $=, <$

one-place function symbol: S

two-place function symbols: $+, \times$

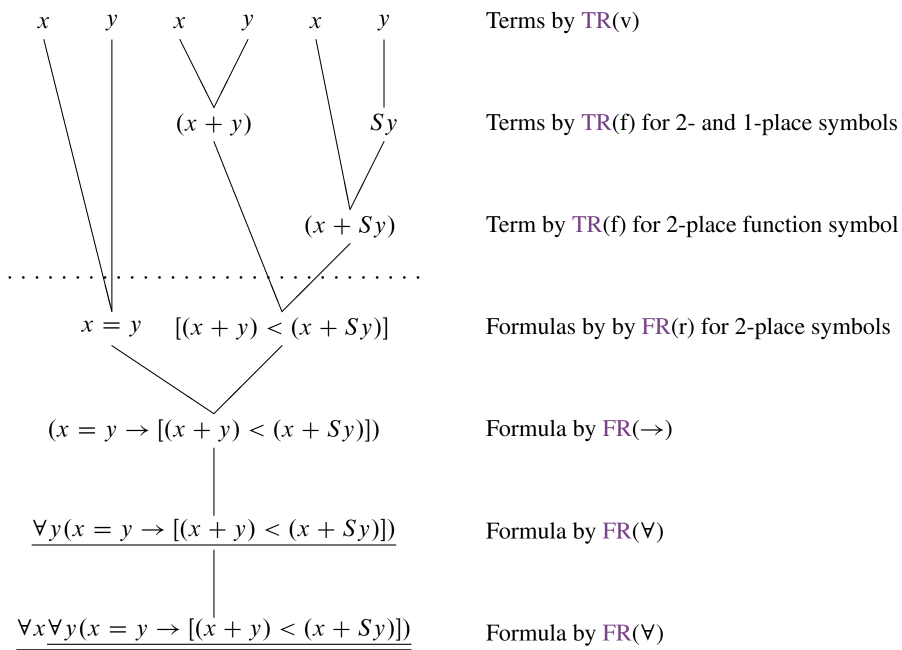
and that is all. Later we shall introduce a language like $\mathscr{L}_{NT}^{<}$ except without the $<$ symbol; for now, we leave it in. Notice that $\mathscr{L}_q$ uses capitals for sentence letters and lowercase for function symbols. But there is nothing sacred about this. Similarly, $\mathscr{L}_q$ indicates the number of places for function and relation symbols by superscripts, where in $\mathscr{L}_{NT}^{<}$ the number of places is simply built into the definition of the symbol. In fact, $\mathscr{L}_{NT}^{<}$ is an extremely simple language! Given the vocabulary, TR and FR apply in the usual way. Thus $\emptyset$, $S\emptyset$ and $SS\emptyset$ are terms — as is easy to see on a tree. And $<\emptyset SS\emptyset$ is an atomic formula.

As with our treatment for equality, for terms m and n, we often abbreviate official terms of the sort, $+mn$ and $\times mn$ as $(m + n)$ and $(m \times n)$; similarly, it is often convenient to abbreviate an atomic formula $<mn$ as $(m < n)$. And we will drop these parentheses when there is no danger of confusion. Officially, we have not said a word about what these expressions mean. It is natural, however, to think of them with their usual meanings, with S the *successor* function — so that the successor of

zero, $S\emptyset$ is one, the successor of the successor of zero $SS\emptyset$ is two, and so forth. But we do not need to think about that for now.

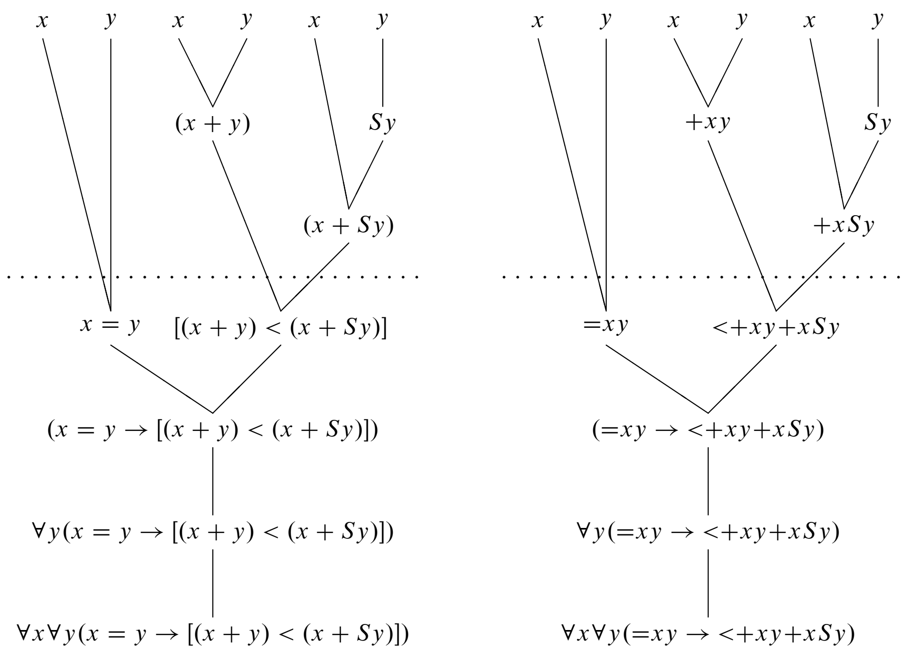
As an example, we show that $\forall x \forall y(x = y \to [(x + y) < (x + Sy)])$ is a(n abbreviation of) a formula and a sentence.

(W)



And we can show what it abbreviates by unpacking the abbreviation in the usual way. This time, we need to pay attention to abbreviations in the terms as well as formulas.

(X)



The official (Polish) notation on the right may seem strange. But it follows the official definitions TR and FR. And it conveniently reduces the number of parentheses from the more typical infix presentation. (You may also be familiar with Polish notation for math from certain electronic calculators.) If you are comfortable with grammar and abbreviations for this language $\mathscr{L}_{NT}^{<}$, you are doing well with the grammar for our formal languages.

E2.21. For each of the following expressions, (i) Demonstrate that it is a formula of $\mathscr{L}_{NT}^{<}$ with a tree. (ii) On the tree bracket all the subformulas, box the immediate subformulas, star the atomic subformulas, circle the main operator, and indicate quantifier scope with underlines. Then (iii) say whether the formula is a sentence, and if it is not, explain why.

a. $\sim[S\emptyset = (S\emptyset \times SS\emptyset)]$

*b. $\exists x \forall y (x \times y = x)$

c. $\forall x[\sim(x = \emptyset) \rightarrow \exists y(y < x)]$

d. $\forall y[(x < y \vee x = y) \vee y < x]$

e. $\forall x \forall y \forall z[(x \times (y + z)) = ((x \times y) + (x \times z))]$

*E2.22. For each of the formulas in E2.21, produce an unabbreviating tree to find the unabbreviated expression it represents.

E2.23. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The vocabulary for a quantificational language and then for $\mathscr{L}_q$ and $\mathscr{L}_{NT}^{<}$.

b. A formula and a sentence of a quantificational language.

c. An abbreviation for an official formula and sentence of a quantificational language.

Chapter 3

Axiomatic Deduction

We have not yet said what our sentences mean. This is just what we do in the next chapter. However, just as it is possible to do grammar without reference to meaning, so it is possible to do derivations without reference to meaning. Derivations are *defined* purely in relation to formula and sentence form. That is why it is crucial to *show* that derivations stand in important relations to validity and truth, as we do in Part III. And that is why it is possible to do derivations without knowing what the expressions mean. In this chapter we develop an *axiomatic* derivation system without any reference to meaning and truth. Apart from relations to meaning and truth, derivations are perfectly well-defined — counting at least as a sort of puzzle or game with, perhaps, a related "thrill of victory" and "agony of defeat." And as with a game, it is possible to build derivation skills, to become a better player. Later, we will show how derivation games matter.[1]

Derivation systems are constructed for different purposes. Introductions to mathematical logic typically employ an *axiomatic* approach. We will see a *natural deduction* system in chapter 6. The advantage of axiomatic systems is their extreme simplicity. From a practical point of view, when we want to think *about* logic, it is convenient to have a relatively simple object to think about. The axiomatic approach makes it natural to build toward increasingly complex and powerful results. As we will see, however, in the beginning, axiomatic derivations can be relatively challenging! We will introduce our system in stages: After some general remarks about what an axiom system is supposed to be, we will introduce the sentential component of

[1]This chapter is out of place. Having developed the grammar of our formal languages, a sensible course in mathematical logic will skip directly to chapter 4 and return only after chapter 6. This chapter has its location to crystallize the the point about form. One might reasonably attempt the first section, but then return only after background from chapters that follow.

our system — the part with application to forms involving just $\sim$ and $\rightarrow$ (and so $\vee$, $\wedge$, and $\leftrightarrow$). After that, we will turn to the full system for forms with quantifiers and equality, including a mathematical application.

3.1 General

Before turning to the derivations themselves, it will be helpful to make some points about the metalanguage and form. First, we are familiar with the idea that different formulas may be of the same form. Thus, for example, where $\mathcal{P}$ and $\mathcal{Q}$ are formulas, $A \rightarrow B$ and $A \rightarrow (B \vee C)$ are both of the form, $\mathcal{P} \rightarrow \mathcal{Q}$ — in the one case $\mathcal{Q}$ maps to B, and in the other to $(B \vee C)$. And, more generally, for formulas $\mathcal{A}, \mathcal{B}, \mathcal{C}$, any formula of the form $\mathcal{A} \rightarrow (\mathcal{B} \vee \mathcal{C})$ is also of the form $\mathcal{P} \rightarrow \mathcal{Q}$. For if $(\mathcal{B} \vee \mathcal{C})$ maps onto some formula, $\mathcal{Q}$ maps onto that formula as well. Of course, this does not go the other way around: it is not the case that every expression of the form $\mathcal{P} \rightarrow \mathcal{Q}$ is of the form $\mathcal{A} \rightarrow (\mathcal{B} \vee \mathcal{C})$; for it is not the case that $\mathcal{B} \vee \mathcal{C}$ maps to any expression to onto which $\mathcal{Q}$ maps. Be sure you are clear about this! Using the metalanguage this way, we can speak generally about formulas in arbitrary sentential or quantificational languages. This is just what we will do — on the assumption that our script letters $\mathcal{A} \ldots \mathcal{Z}$ range over formulas of some arbitrary formal language $\mathcal{L}$, we frequently depend on the fact that every formula of one form is also of another.

Given a formal language $\mathcal{L}$, an axiomatic logic *AL* consists of two parts. There is a set of *axioms* and a set of *rules*. Different axiomatic logics result from different axioms and rules. For now, the set of axioms is just some privileged collection of formulas. A rule tells us that one formula *follows* from some others. One way to specify axioms and rules is by form. Thus, for example, *modus ponens* may be included among the rules.

MP $\quad \mathcal{P} \rightarrow \mathcal{Q}, \mathcal{P} \vdash \mathcal{Q}$

The $\vdash$ symbol is *single turnstile* (to contrast with a *double* turnstile $\vDash$ from chapter 4). According to this rule, for any formulas $\mathcal{P}$ and $\mathcal{Q}$, the formula $\mathcal{Q}$ *follows* from $\mathcal{P} \rightarrow \mathcal{Q}$ together with $\mathcal{P}$. Thus, as applied to $\mathcal{L}_s$, B follows by MP from $A \rightarrow B$ and A; but also $(B \leftrightarrow D)$ follows from $(A \rightarrow B) \rightarrow (B \leftrightarrow D)$ and $(A \rightarrow B)$. And for a case put in the metalanguage, quite generally, a formula of the form $(\mathcal{A} \wedge \mathcal{B})$ follows from $\mathcal{A} \rightarrow (\mathcal{A} \wedge \mathcal{B})$ and $\mathcal{A}$ — for any formulas of the form $\mathcal{A} \rightarrow (\mathcal{A} \wedge \mathcal{B})$ and $\mathcal{A}$ are of the forms $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ as well. Axioms also may be specified by form. Thus, for some language with formulas $\mathcal{P}$ and $\mathcal{Q}$, a logic might include all formulas of the forms,

$\wedge 1 \quad (\mathcal{P} \wedge \mathcal{Q}) \rightarrow \mathcal{P} \qquad \wedge 2 \quad (\mathcal{P} \wedge \mathcal{Q}) \rightarrow \mathcal{Q} \qquad \wedge 3 \quad \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow (\mathcal{P} \wedge \mathcal{Q}))$

among its axioms. Then in $\mathcal{L}_s$,

$$(A \wedge B) \rightarrow A, \qquad (A \wedge A) \rightarrow A \qquad ((A \rightarrow B) \wedge C) \rightarrow (A \rightarrow B)$$

are all axioms of form $\wedge 1$. So far, for a given axiomatic logic *AL*, there are no constraints on just which forms will be the axioms, and just which rules are included. The point is only that we specify an axiomatic logic when we specify some collection of axioms and rules.

Suppose we have specified some axioms and rules for an axiomatic logic *AL*. Then where Γ (Gamma), is a set of formulas — taken as the formal *premises* of an argument,

AV (p) If $\mathcal{P}$ is a premise (a member of Γ), then $\mathcal{P}$ is a *consequence* in *AL* of Γ.

(a) If $\mathcal{P}$ is an axiom of *AL*, then $\mathcal{P}$ is a *consequence* in *AL* of Γ.

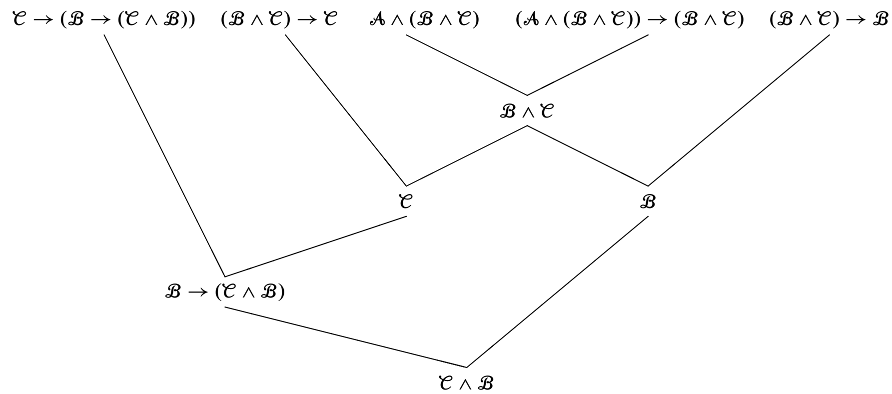
(r) If $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ are consequences in *AL* of Γ, and there is a rule of *AL* such that $\mathcal{P}$ follows from $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ by the rule, then $\mathcal{P}$ is a *consequence* in *AL* of Γ.

(CL) Any *consequence* in *AL* of Γ may be obtained by repeated application of these rules.

The first two clauses make premises and axioms consequences in *AL* of Γ. And if, say, MP is a rule of an *AL* and $P \rightarrow Q$ and P are consequences in *AL* of Γ, then by AV(r), Q is a consequence in *AL* of Γ as well. If $\mathcal{P}$ is a consequence in *AL* of some premises Γ, then the premises *prove* $\mathcal{P}$ in *AL* and we write $\Gamma \vdash_{AL} \mathcal{P}$; in this case the argument is *valid* in *AL*. If $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ are the members of Γ, we sometimes write $\mathcal{Q}_1 \ldots \mathcal{Q}_n \vdash_{AL} \mathcal{P}$ in place of $\Gamma \vdash_{AL} \mathcal{P}$. If Γ has no members at all and $\Gamma \vdash_{AL} \mathcal{P}$, then $\mathcal{P}$ is a *theorem* of *AL*. In this case, listing all the premises individually, we simply write, $\vdash_{AL} \mathcal{P}$.

Before turning to our official axiomatic system *AD*, it will be helpful to consider a simple example. Suppose an axiomatic derivation system *A1* has MP as its only rule, and just formulas of the forms $\wedge 1$, $\wedge 2$, and $\wedge 3$ as axioms. AV is a recursive definition like ones we have seen before. Thus nothing stops us from working out its consequences on trees. Thus we can show that $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C}) \vdash_{A1} \mathcal{C} \wedge \mathcal{B}$ as follows,

(A)



For definition AV, the basic elements are the premises and axioms. These occur across the top row. Thus, reading from the left, the first form is an instance of $\wedge 3$. The second is of type $\wedge 2$. These are thus consequences of Γ by AV(a). The third is the premise. Thus it is a consequence by AV(p). Any formula of the form $(\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})) \to (\mathcal{B} \wedge \mathcal{C})$ is of the form, $(\mathcal{P} \wedge \mathcal{Q}) \to \mathcal{Q}$; so the fourth is of the type $\wedge 2$. And the last is of the type $\wedge 1$. So the final two are consequences by AV(a). After that, all the results are by MP, and so consequences by AV(r). Thus for example, in the first case, $(\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})) \to (\mathcal{B} \wedge \mathcal{C})$ and $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ are of the sort $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$, with $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ for $\mathcal{P}$ and $(\mathcal{B} \wedge \mathcal{C})$ for $\mathcal{Q}$; thus $\mathcal{B} \wedge \mathcal{C}$ follows from them by MP. So $\mathcal{B} \wedge \mathcal{C}$ is a consequence in *A1* of Γ by AV(r). And similarly for the other consequences. Notice that applications of MP and of the axiom forms are *independent* from one use to the next. The expressions that count as $\mathcal{P}$ or $\mathcal{Q}$ must be consistent within a given application of the axiom or rule, but may vary from one application of the axiom or rule to the next. If you are familiar with another derivation system, perhaps the one from chapter 6, you may think of an axiom as a rule without inputs. Then the axiom applies to expressions of its form in the usual way.

These diagrams can get messy, and it is traditional to represent the same information as follows, using annotations to indicate relations among formulas.

(B)

1.	$\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$	prem(ise)
2.	$(\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})) \to (\mathcal{B} \wedge \mathcal{C})$	$\wedge 2$
3.	$\mathcal{B} \wedge \mathcal{C}$	2,1 MP
4.	$(\mathcal{B} \wedge \mathcal{C}) \to \mathcal{B}$	$\wedge 1$
5.	$\mathcal{B}$	4,3 MP
6.	$(\mathcal{B} \wedge \mathcal{C}) \to \mathcal{C}$	$\wedge 2$
7.	$\mathcal{C}$	6,3 MP
8.	$\mathcal{C} \to (\mathcal{B} \to (\mathcal{C} \wedge \mathcal{B}))$	$\wedge 3$
9.	$\mathcal{B} \to (\mathcal{C} \wedge \mathcal{B})$	8,7 MP
10.	$\mathcal{C} \wedge \mathcal{B}$	9,5 MP

Each of the forms (1) - (10) is a consequence of $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ in *A1*. As indicated on the right, the first is a premise, and so a consequence by AV(p). The second is an axiom of the form $\wedge$2, and so a consequence by AV(a). The third follows by MP from the forms on lines (2) and (1), and so is a consequence by AV(r). And so forth. Such a demonstration is an *axiomatic derivation*. This derivation contains the very same information as the tree diagram (A), only with geometric arrangement replaced by line numbers to indicate relations between forms. Observe that we might have accomplished the same end with a different arrangement of lines. For example, we might have listed all the axioms first, with applications of MP after. The important point is that in an *axiomatic derivation*, each line is either an axiom, a premise, or follows from previous lines by a rule. Just as a tree is sufficient to demonstrate that $\Gamma \vdash_{AL} \mathcal{P}$, that $\mathcal{P}$ is a consequence of Γ in *AL*, so an axiomatic derivation is sufficient to show the same. In fact, we shall typically use derivations, rather than trees to show that $\Gamma \vdash_{AL} \mathcal{P}$.

Notice that we have been reasoning with sentence *forms*, and so have shown that a formula of the form $\mathcal{C} \wedge \mathcal{B}$ follows in *A1* from one of the form $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$. Given this, we freely appeal to results of one derivation in the process of doing another. Thus, if we were to encounter a formula of the form $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$ in an *A1* derivation, we might simply cite the derivation (B) completed above, and move directly to the conclusion that $\mathcal{C} \wedge \mathcal{B}$. The resultant derivation would be an *abbreviation* of an official one which includes each of the above steps to reach $\mathcal{C} \wedge \mathcal{B}$. In this way, derivations remain manageable, and we are able to build toward results of increasing complexity. (Compare your high school experience of Euclidian geometry.) All of this should become more clear, as we turn to the official and complete axiomatic system, *AD*.

Again, unless you have a special reason for studying axiomatic systems, or are just looking for some really challenging puzzles, you should move on to the next chapter after these exercises and return only after chapter 6. This chapter makes sense here for conceptual reasons, but is completely out of order from a learning point of view. After chapter 6 you can return to this chapter, but recognize its place in in the conceptual order.

E3.1. Where *A1* is as above with rule MP and axioms $\wedge$1-3, construct derivations to show each of the following.

*a. $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C}) \vdash_{A1} \mathcal{B}$

b. $\mathcal{A}, \mathcal{B}, \mathcal{C} \vdash_{A1} \mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})$

c. $\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C}) \vdash_{A1} (\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}$

d. $(\mathcal{A} \wedge \mathcal{B}) \wedge (\mathcal{C} \wedge \mathcal{D}) \vdash_{A1} \mathcal{B} \wedge \mathcal{C}$

e. $\vdash_{A1} ((\mathcal{A} \wedge \mathcal{B}) \to \mathcal{A}) \wedge ((\mathcal{A} \wedge \mathcal{B}) \to \mathcal{B})$

3.2 Sentential

We begin by focusing on sentential forms, forms involving just $\sim$ and $\to$ (and so $\wedge$, $\vee$ and $\leftrightarrow$). The sentential component of our official axiomatic logic *AD* tells us how to manipulate such forms, whether they be forms for expressions in a sentential language like $\mathcal{L}_s$, or in a quantificational language like $\mathcal{L}_q$. The sentential fragment of *AD* includes three forms for logical axioms, and one rule.

AS A1. $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$

A2. $(\mathcal{O} \to (\mathcal{P} \to \mathcal{Q})) \to ((\mathcal{O} \to \mathcal{P}) \to (\mathcal{O} \to \mathcal{Q}))$

A3. $(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$

MP $\mathcal{P} \to \mathcal{Q}, \mathcal{P} \vdash \mathcal{Q}$

We have already encountered MP. To take some cases to appear immediately below, the following are both of the sort A1.

$$\mathcal{A} \to (\mathcal{A} \to \mathcal{A}) \qquad (\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$$

Observe that $\mathcal{P}$ and $\mathcal{Q}$ need not be different! You should be clear about these cases. Although MP is the only rule, we allow free movement between an expression and its abbreviated forms, with justification, "abv." That is it! As above, $\Gamma \vdash_{AD} \mathcal{P}$ just in case $\mathcal{P}$ is a consequence of Γ in *AD*. $\Gamma \vdash_{AD} \mathcal{P}$ just in case there is a derivation of $\mathcal{P}$ from premises in Γ.

The following is a series of derivations where, as we shall see, each may depend on ones from before. At first, do not worry so much about strategy, as about the mechanics of the system.

T3.1. $\vdash_{AD} \mathcal{A} \to \mathcal{A}$

1.	$\mathcal{A} \to ([\mathcal{A} \to \mathcal{A}] \to \mathcal{A})$	A1
2.	$(\mathcal{A} \to ([\mathcal{A} \to \mathcal{A}] \to \mathcal{A})) \to ((\mathcal{A} \to [\mathcal{A} \to \mathcal{A}]) \to (\mathcal{A} \to \mathcal{A}))$	A2
3.	$(\mathcal{A} \to [\mathcal{A} \to \mathcal{A}]) \to (\mathcal{A} \to \mathcal{A})$	2,1 MP
4.	$\mathcal{A} \to [\mathcal{A} \to \mathcal{A}]$	A1
5.	$\mathcal{A} \to \mathcal{A}$	3,4 MP

Line (1) is an axiom of the form A1 with $\mathcal{A} \to \mathcal{A}$ for $\mathcal{Q}$. Line (2) is an axiom of the form A2 with $\mathcal{A}$ for $\mathcal{O}$, $\mathcal{A} \to \mathcal{A}$ for $\mathcal{P}$, and $\mathcal{A}$ for $\mathcal{Q}$. Notice again that $\mathcal{O}$ and $\mathcal{Q}$ may be any formulas, so nothing prevents them from being the same. Similarly, line (4) is an axiom of form A1 with $\mathcal{A}$ in place of both $\mathcal{P}$ and $\mathcal{Q}$. The applications of MP should be straightforward.

T3.2. $\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C} \vdash_{AD} \mathcal{A} \to \mathcal{C}$

1.	$\mathcal{B} \to \mathcal{C}$	prem
2.	$(\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$	A1
3.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2,1 MP
4.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
5.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	4,3 MP
6.	$\mathcal{A} \to \mathcal{B}$	prem
7.	$\mathcal{A} \to \mathcal{C}$	5,6 MP

Line (4) is an instance of A2 which gives us our goal with two applications of MP — that is, from (4), $\mathcal{A} \to \mathcal{C}$ follows by MP if we have $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$ and $\mathcal{A} \to \mathcal{B}$. But the second of these is a premise, so the only real challenge is getting $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$. But since $\mathcal{B} \to \mathcal{C}$ is a premise, we can use A1 to get *anything* arrow it — and that is just what we do by the first three lines.

T3.3. $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{AD} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$

1.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$	A1
2.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	prem
3.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
4.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	3,2 MP
5.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$	1,4 T3.2

In this case, the first four steps are very much like ones you have seen before. But the last is not. We have $\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$ on line (1), and $(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$ on line (4). These are of the form to be inputs to T3.2 — with $\mathcal{B}$ for $\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$ for $\mathcal{B}$, and $\mathcal{A} \to \mathcal{C}$ for $\mathcal{C}$. T3.2 is a sort of transitivity or "chain" principle which lets us move from a first form to a last through some middle term. In this case, $\mathcal{A} \to \mathcal{B}$ is the middle term. So at line (5), we simply observe that lines (1) and (4), together with the reasoning from T3.2, give us the desired result.

What we have not produced is an official derivation, where each step is a premise, an axiom, or follows from previous lines by a rule. But we have produced an abbreviation of one. And nothing prevents us from unabbreviating by including the routine from T3.2 to produce a derivation in the official form. To see this, first, observe

that the derivation for T3.2 has its premises at lines (1) and (6), where lines with the corresponding forms in the derivation for T3.3 appear at (4) and (1). However, it is a simple matter to reorder the derivation for T3.2 so that it takes its premises from those same lines. Thus here is another demonstration for T3.2.

(C)

1.	$\mathcal{A} \to \mathcal{B}$	prem
⋮		
4.	$\mathcal{B} \to \mathcal{C}$	prem
5.	$(\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$	A1
6.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	5,4 MP
7.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
8.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	7,6 MP
9.	$\mathcal{A} \to \mathcal{C}$	8,1 MP

Compared to the original derivation for T3.2, all that is different is the order of a few lines, and corresponding line numbers. The *reason* for reordering the lines is for a merge of this derivation with the one for T3.3.

But now, although we are after expressions of the *form* $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{C}$, the actual expressions we want for T3.3 are $\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$ and $(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$. But we can convert derivation (C) to one with those very forms by uniform substituation of $\mathcal{B}$ for every $\mathcal{A}$; $(\mathcal{A} \to \mathcal{B})$ for every $\mathcal{B}$; and $(\mathcal{A} \to \mathcal{C})$ for every $\mathcal{C}$ — that is, we apply our original map to the entire derivation (C). The result is as follows.

(D)

1.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$	prem
⋮		
4.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	prem
5.	$((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})) \to [\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))]$	A1
6.	$\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))$	5,4 MP
7.	$[\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))] \to [(\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C}))]$	A2
8.	$(\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C}))$	7,6 MP
9.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$	8,1 MP

You should trace the parallel between derivations (C) and (D) all the way through. And you should verify that (D) is a derivation on its own. This is an application of the point that our derivation for T3.2 applies to any premises and conclusions of that form. The result is a direct demonstration that $\mathcal{B} \to (\mathcal{A} \to \mathcal{B}), (\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}) \vdash_{AD} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$.

And now it is a simple matter to merge the lines from (D) into the derivation for T3.3 to produce a complete demonstration that $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{AD} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$.

(E)

1.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{B})$	A1
2.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	prem
3.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
4.	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	3,2 MP
5.	$((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})) \to [\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))]$	A1
6.	$\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))$	5,4 MP
7.	$[\mathcal{B} \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))] \to [(\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C}))]$	A2
8.	$(\mathcal{B} \to (\mathcal{A} \to \mathcal{B})) \to (\mathcal{B} \to (\mathcal{A} \to \mathcal{C}))$	7,6 MP
9.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$	8,1 MP

Lines (1) - (4) are the same as from the derivation for T3.3, and include what are the premises to (D). Lines (5) - (9) are the same as from (D). The result is a demonstration for T3.3 in which every line is a premise, an axiom, or follows from previous lines by MP. Again, you should follow each step. It is hard to believe that we could *think up* this last derivation — particularly at this early stage of our career. However, if we can produce the simpler derivation, we can be sure that this more complex one exists. Thus we can be sure that the final result is a consequence of the premise in *AD*. That is the point of our direct appeal to T3.2 in the original derivation of T3.3. And similarly in cases that follow. In general, we are always free to appeal to prior results in any derivation — so that our toolbox gets bigger at every stage.

T3.4. $\vdash_{AD} (\mathcal{B} \to \mathcal{C}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$

1.	$(\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$	A1
2.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	A2
3.	$(\mathcal{B} \to \mathcal{C}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	1,2 T3.2

Again, we have an application of T3.2. In this case, the middle term (the $\mathcal{B}$) from T3.2 maps to $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$. Once we see that the consequent of what we want is like the consequent of A2, we should be "inspired" by T3.2 to go for (1) as a link between the antecedent of what we want, and antecedent of A2. As it turns out, this is easy to get as an instance of A1. It is helpful to say to yourself in words, what the various axioms and theorems do. Thus, given some $\mathcal{P}$, A1 yields *anything* arrow it. And T3.2 is a simple transitivity principle.

T3.5. $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to [(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to \mathcal{C})]$

1.	$(\mathcal{B} \to \mathcal{C}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$	T3.4
2.	$(\mathcal{A} \to \mathcal{B}) \to [(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to \mathcal{C})]$	1 T3.3

T3.5 is like T3.4 except that $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{C}$ switch places. But T3.3 precisely switches terms in those places — with $\mathcal{B} \to \mathcal{C}$ for $\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$ for $\mathcal{B}$, and $\mathcal{A} \to \mathcal{C}$

for $\mathscr{C}$. Again, often what is difficult about these derivations is "seeing" what you can do. Thus it is good to say to yourself in words what the different principles give you. Once you realize what T3.3 does, it is obvious that you have T3.5 immediately from T3.4.

T3.6. $\mathscr{B}, \mathscr{A} \rightarrow (\mathscr{B} \rightarrow \mathscr{C}) \vdash_{AD} \mathscr{A} \rightarrow \mathscr{C}$

Hint: You can get this in the basic system using just A1 and A2. But you can get it in just four lines if you use T3.3.

T3.7. $\vdash_{AD} ({\sim}\mathscr{A} \rightarrow \mathscr{A}) \rightarrow \mathscr{A}$

Hint: This follows in just three lines from A3, with an instance of T3.1.

T3.8. $\vdash_{AD} ({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow (\mathscr{A} \rightarrow \mathscr{B})$

1.	$({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow [({\sim}\mathscr{B} \rightarrow \mathscr{A}) \rightarrow \mathscr{B}]$	A3
2.	$[({\sim}\mathscr{B} \rightarrow \mathscr{A}) \rightarrow \mathscr{B}] \rightarrow [(\mathscr{A} \rightarrow ({\sim}\mathscr{B} \rightarrow \mathscr{A})) \rightarrow (\mathscr{A} \rightarrow \mathscr{B})]$	T3.4
3.	$\mathscr{A} \rightarrow ({\sim}\mathscr{B} \rightarrow \mathscr{A})$	A1
4.	$[({\sim}\mathscr{B} \rightarrow \mathscr{A}) \rightarrow \mathscr{B}] \rightarrow (\mathscr{A} \rightarrow \mathscr{B})$	2,3 T3.6
5.	$({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow (\mathscr{A} \rightarrow \mathscr{B})$	1,4 T3.2

The idea behind this derivation is that the antecedent of A3 is the antecedent of our goal. So we can get the goal by T3.2 with the instance of A3 on (1) and (4). That is, given $({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow \mathscr{X}$, what we need to get the goal by an application of T3.2 is $\mathscr{X} \rightarrow (\mathscr{A} \rightarrow \mathscr{B})$. But that is just what (4) is. The challenge is to get (4). Our strategy uses T3.4, and then T3.6 with A1 to "delete" the middle term. This derivation is not particularly easy to see. Here is another approach, which is not all that easy either.

(F)

1.	$({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow [({\sim}\mathscr{B} \rightarrow \mathscr{A}) \rightarrow \mathscr{B}]$	A3
2.	$({\sim}\mathscr{B} \rightarrow \mathscr{A}) \rightarrow [({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow \mathscr{B}]$	1 T3.3
3.	$\mathscr{A} \rightarrow ({\sim}\mathscr{B} \rightarrow \mathscr{A})$	A1
4.	$\mathscr{A} \rightarrow [({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow \mathscr{B}]$	3,2 T3.2
5.	$({\sim}\mathscr{B} \rightarrow {\sim}\mathscr{A}) \rightarrow (\mathscr{A} \rightarrow \mathscr{B})$	4 T3.3

This derivation also begins with A3. The idea this time is to use T3.3 to "swing" ${\sim}\mathscr{B} \rightarrow \mathscr{A}$ out, "replace" it by $\mathscr{A}$ with T3.2 and A1, and then use T3.3 to "swing" $\mathscr{A}$ back in.

T3.9. $\vdash_{AD} {\sim}\mathscr{A} \rightarrow (\mathscr{A} \rightarrow \mathscr{B})$

Hint: You can do this in three lines with T3.8 and an instance of A1.

T3.10. $\vdash_{AD} \sim\sim\mathcal{A} \to \mathcal{A}$

Hint: You can do this in three lines wih instances of T3.7 and T3.9.

T3.11. $\vdash_{AD} \mathcal{A} \to \sim\sim\mathcal{A}$

Hint: You can do this in three lines with instances of T3.8 and T3.10.

*T3.12. $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to (\sim\sim\mathcal{A} \to \sim\sim\mathcal{B})$

Hint: Use T3.5 and T3.10 to get $(\mathcal{A} \to \mathcal{B}) \to (\sim\sim\mathcal{A} \to \mathcal{B})$; then use T3.4, and T3.11 to get $(\sim\sim\mathcal{A} \to \mathcal{B}) \to (\sim\sim\mathcal{A} \to \sim\sim\mathcal{B})$; the result follows easily by T3.2.

T3.13. $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to (\sim\mathcal{B} \to \sim\mathcal{A})$

Hint: You can do this in three lines with instances of T3.8 and T3.12.

T3.14. $\vdash_{AD} (\sim\mathcal{A} \to \mathcal{B}) \to (\sim\mathcal{B} \to \mathcal{A})$

Hint: Use T3.4 and T3.10 to get $(\sim\mathcal{B} \to \sim\sim\mathcal{A}) \to (\sim\mathcal{B} \to \mathcal{A})$; the result follows easily with an instance of T3.13.

T3.15. $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to [(\sim\mathcal{A} \to \mathcal{B}) \to \mathcal{B}]$

Hint: Use T3.13 and A3 to get $(\mathcal{A} \to \mathcal{B}) \to [(\sim\mathcal{B} \to \mathcal{A}) \to \mathcal{B}]$; then use T3.5 and T3.14 to get $[(\sim\mathcal{B} \to \mathcal{A}) \to \mathcal{B}] \to [(\sim\mathcal{A} \to \mathcal{B}) \to \mathcal{B}]$; the result follows easily by T3.2.

*T3.16. $\vdash_{AD} \mathcal{A} \to [\sim\mathcal{B} \to \sim(\mathcal{A} \to \mathcal{B})]$

Hint: Use instances of T3.1 and T3.3 to get $\mathcal{A} \to [(\mathcal{A} \to \mathcal{B}) \to \mathcal{B}]$; then use T3.13 to "turn around" the consequent. This idea of deriving conditionals in "reversed" form, and then using T3.13 or T3.14 to turn them around, is frequently useful for getting tilde outside of a complex expression.

T3.17. $\vdash_{AD} \mathcal{A} \to (\mathcal{A} \vee \mathcal{B})$

1.	$\sim\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	T3.9
2.	$\mathcal{A} \to (\sim\mathcal{A} \to \mathcal{B})$	1 T3.3
3.	$\mathcal{A} \to (\mathcal{A} \vee \mathcal{B})$	2 abv

We set as our goal the unabbreviated form. We have this at (2). Then, in the last line, simply observe that the goal abbreviates what has already been shown.

T3.18. $\vdash_{AD} \mathscr{A} \to (\mathscr{B} \vee \mathscr{A})$

Hint: Go for $\mathscr{A} \to (\sim\mathscr{B} \to \mathscr{A})$. Then, as above, you can get the desired result in one step by abv.

T3.19. $\vdash_{AD} (\mathscr{A} \wedge \mathscr{B}) \to \mathscr{B}$

T3.20. $\vdash_{AD} (\mathscr{A} \wedge \mathscr{B}) \to \mathscr{A}$

*T3.21. $\mathscr{A} \to (\mathscr{B} \to \mathscr{C}) \vdash_{AD} (\mathscr{A} \wedge \mathscr{B}) \to \mathscr{C}$

T3.22. $(\mathscr{A} \wedge \mathscr{B}) \to \mathscr{C} \vdash_{AD} \mathscr{A} \to (\mathscr{B} \to \mathscr{C})$

T3.23. $\mathscr{A}, \mathscr{A} \leftrightarrow \mathscr{B} \vdash_{AD} \mathscr{B}$

Hint: $\mathscr{A} \leftrightarrow \mathscr{B}$ abbreviates the same thing as $(\mathscr{A} \to \mathscr{B}) \wedge (\mathscr{B} \to \mathscr{A})$; you may thus move to this expression from $\mathscr{A} \leftrightarrow \mathscr{B}$ by abv.

T3.24. $\mathscr{B}, \mathscr{A} \leftrightarrow \mathscr{B} \vdash_{AD} \mathscr{A}$

T3.25. $\sim\mathscr{A}, \mathscr{A} \leftrightarrow \mathscr{B} \vdash_{AD} \sim\mathscr{B}$

T3.26. $\sim\mathscr{B}, \mathscr{A} \leftrightarrow \mathscr{B} \vdash_{AD} \sim\mathscr{A}$

*E3.2. Provide derivations for T3.6, T3.7, T3.9, T3.10, T3.11, T3.12, T3.13, T3.14, T3.15, T3.16, T3.18, T3.19, T3.20, T3.21, T3.22, T3.23, T3.24, T3.25, and T3.26. As you are working these problems, you may find it helpful to refer to the *AD* summary on p. 87.

E3.3. For each of the following, expand derivations to include all the steps from theorems. The result should be a derivation in which each step is either a premise, an axiom, or follows from previous lines by a rule. Hint: it may be helpful to proceed in stages as for (C), (D) and then (E) above.

a. Expand your derivation for T3.7.

*b. Expand the above derivation for T3.4.

E3.4. Consider an axiomatic system *A2* which takes $\wedge$ and $\sim$ as primitive operators, and treats $\mathcal{P} \rightarrow \mathcal{Q}$ as an abbreviation for $\sim(\mathcal{P} \wedge \sim\mathcal{Q})$. The axiom schemes are,

A2 A1. $\mathcal{P} \rightarrow (\mathcal{P} \wedge \mathcal{P})$

A2. $(\mathcal{P} \wedge \mathcal{Q}) \rightarrow \mathcal{P}$

A3. $(\mathcal{O} \rightarrow \mathcal{P}) \rightarrow [\sim(\mathcal{P} \wedge \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \wedge \mathcal{O})]$

MP is the only rule. Provide derivations for each of the following, where derivations may appeal to any *prior* result (no matter what *you* have done).

*a. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C} \vdash_{A2} \sim(\sim\mathcal{C} \wedge \mathcal{A})$

b. $\vdash_{A2} \sim(\sim\mathcal{A} \wedge \mathcal{A})$

c. $\vdash_{A2} \sim\sim\mathcal{A} \rightarrow \mathcal{A}$

*d. $\vdash_{A2} \sim(\mathcal{A} \wedge \mathcal{B}) \rightarrow (\mathcal{B} \rightarrow \sim\mathcal{A})$

e. $\vdash_{A2} \mathcal{A} \rightarrow \sim\sim\mathcal{A}$

f. $\vdash_{A2} (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\sim\mathcal{B} \rightarrow \sim\mathcal{A})$

*g. $\sim\mathcal{A} \rightarrow \sim\mathcal{B} \vdash_{A2} \mathcal{B} \rightarrow \mathcal{A}$

h. $\mathcal{A} \rightarrow \mathcal{B} \vdash_{A2} (\mathcal{C} \wedge \mathcal{A}) \rightarrow (\mathcal{B} \wedge \mathcal{C})$

*i. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C}, \mathcal{C} \rightarrow \mathcal{D} \vdash_{A2} \mathcal{A} \rightarrow \mathcal{D}$

j. $\vdash_{A2} \mathcal{A} \rightarrow \mathcal{A}$

k. $\vdash_{A2} (\mathcal{A} \wedge \mathcal{B}) \rightarrow (\mathcal{B} \wedge \mathcal{A})$

l. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C} \vdash_{A2} \mathcal{A} \rightarrow \mathcal{C}$

m. $\sim\mathcal{B} \rightarrow \mathcal{B} \vdash_{A2} \mathcal{B}$

n. $\mathcal{B} \rightarrow \sim\mathcal{B} \vdash_{A2} \sim\mathcal{B}$

o. $\vdash_{A2} (\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{B}$

p. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{C} \rightarrow \mathcal{D} \vdash_{A2} (\mathcal{A} \wedge \mathcal{C}) \rightarrow (\mathcal{B} \wedge \mathcal{D})$

q. $\mathcal{B} \rightarrow \mathcal{C} \vdash_{A2} (\mathcal{A} \wedge \mathcal{B}) \rightarrow (\mathcal{A} \wedge \mathcal{C})$

r. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{A} \rightarrow \mathcal{C} \vdash_{A2} \mathcal{A} \rightarrow (\mathcal{B} \wedge \mathcal{C})$

s. $\vdash_{A2} [(\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}] \rightarrow [\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})]$

t. $\vdash_{A2} [\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})] \rightarrow [(\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}]$

*u. $\vdash_{A2} [\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})] \rightarrow [(\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{C})]$

v. $\vdash_{A2} [(\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{C}] \rightarrow [\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})]$

*w. $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}) \vdash_{A2} \mathcal{A} \rightarrow \mathcal{C}$

x. $\vdash_{A2} \mathcal{A} \rightarrow [\mathcal{B} \rightarrow (\mathcal{A} \wedge \mathcal{B})]$

y. $\vdash_{A2} \mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{A})$

Hints: *(i)*: Apply (a) to the first two premises and (f) to the third; then recognize that you have the makings for an application of A3. *(j)*: Apply A1, two

instances of (h), and an instance of (i) to get $\mathcal{A} \to ((\mathcal{A} \wedge \mathcal{A}) \wedge (\mathcal{A} \wedge \mathcal{A}))$; the result follows easily with A2 and (i). *(m)*: $\sim\mathcal{B} \to \mathcal{B}$ is equivalent to $\sim(\sim\mathcal{B} \wedge \sim\mathcal{B})$; and $\sim\mathcal{B} \to (\sim\mathcal{B} \wedge \sim\mathcal{B})$ is immediate from A2; you can turn this around by (f) to get $\sim(\sim\mathcal{B} \wedge \sim\mathcal{B}) \to \sim\sim\mathcal{B}$; then it is easy. *(u)*: Use abv so that you are going for $\sim[\mathcal{A} \wedge \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})] \to \sim[(\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C}]$; plan on getting to this by (f); the proof then reduces to working from $((\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C})$. *(v)*: Structure your proof very much as with (u). *(w)*: Use (u) to set up a "chain" to which you can apply transitivity.

3.3 Quantificational

We begin this section by introducing one new rule, and some axioms for quantifier forms. There will be an one axiom and one rule for manipulating quantifiers, and some axioms for features of equality. After introducing the axioms and rule, we use them with application to some theorems of *Peano Arithmetic*.

3.3.1 Quantifiers

Excluding equality, to work with quantifier forms, we add just one axiom form and one rule. To state the axiom, we need a couple of definitions. First, for any formula $\mathcal{A}$, variable x, and term t, say $\mathcal{A}^x_t$ is $\mathcal{A}$ with all the free instances of x replaced by t. And say t is *free for x in $\mathcal{A}$* iff all the variables in the replacing instances of t remain free after substitution in $\mathcal{A}^x_t$. Thus, for example, where $\mathcal{A}$ is $(\forall xRxy \vee Px)$,

(G) $(\forall xRxy \vee Px)^x_y$ is $\forall xRxy \vee Py$

There are three instances of x in $\forall xRxy \vee Px$, but only the last is free; so y is substituted only for that instance. Since the substituted y is free in the resultant expression, y is free for x in $\forall xRxy \vee Px$. Similarly,

(H) $[\forall x(x = y) \vee Ryx]^y_{f^1x}$ is $\forall x(x = f^1x) \vee Rf^1xx$

Both instances of y in $\forall x(x = y) \vee Ryx$ are free; so our substitution replaces both. But the x in the first instance of f^1x is bound upon substitution; so f^1x is not free for y in $\forall x(x = y) \vee Ryx$. Notice that if x is not free in $\mathcal{A}$, then replacing every free instance of x in $\mathcal{A}$ with some term results in no change. So if x is not free in $\mathcal{A}$, then $\mathcal{A}^x_t$ is $\mathcal{A}$. Similarly, $\mathcal{A}^x_x$ is just $\mathcal{A}$ itself. Further, any variable x is sure to be free for itself in a formula $\mathcal{A}$ — if every *free* instance of variable x is "replaced" with x, then the replacing instances are sure to be free! And constants are sure to be free for a variable x in a formula $\mathcal{A}$. Since a constant c is a term without variables, no variable in the replacing term is bound upon substitution for free instances of x.

Now we are ready for our axiom and rule. For the quantificational version of axiomatic derivation system *AD*, in addition to A1, A2, A3 and MP from AS, we add an axiom A4 and a rule Gen (*Generalization*).

AQ A4. $\forall x \mathcal{P} \to \mathcal{P}^x_t$ — where t is free for x in $\mathcal{P}$

Gen $\mathcal{A} \to \mathcal{B} \vdash \mathcal{A} \to \forall x \mathcal{B}$ — where x is not free in $\mathcal{A}$

A1, A2, A3 and MP remain from before. The axiom A4 and rule Gen are new. A4 is a conditional in which the antecedent is a quantified expression; the consequent drops the quantifier, and substitutes term t for each free instance of the quantified variable in the resulting $\mathcal{P}$ — subject to the constraint that the term t is free for the quantified variable in $\mathcal{P}$. Thus, for example in $\mathcal{L}_q$,

(I) $\forall x Rx \to Rx$ $\quad \forall x Rx \to Ry$ $\quad \forall x Rx \to Ra$ $\quad \forall x Rx \to Rf^1 z$

are all instances of A4. In these cases, $\mathcal{P}$ is Rx; x is free in it, and since Rx includes no quantifier, it is easy to see that the substituted terms are all free for x in it. So each of these satisfies the condition on A4. The following are also instances of A4.

(J) $\forall x \forall y Rxy \to \forall y Rzy$ $\quad \forall x \forall y Rxy \to \forall y Rf^1 xy$

In each case, we drop the main quantifier, and substitute a term for the quantified variable, where the substituted term remains free in the resultant expression. However these cases contrast with the ones that follow.

(K) $\forall x \forall y Rxy \to \forall y Ryy$ $\quad \forall x \forall y \to \forall y Rf^1 yy$

In these cases, we drop the quantifier and make a substitution as before. But the substituted terms *are not free*. So the constraint on A4 is violated, and these formulas do not qualify as instances of the axiom.

The new rule also comes with a constraint. Given $\mathcal{P} \to \mathcal{Q}$, one may move to $\mathcal{P} \to \forall x \mathcal{Q}$ *so long as x is not free in $\mathcal{P}$*. Thus the leftmost three cases below are legitimate applications of Gen, where the right-hand case is not.

(L) $\dfrac{Ry \to Sx}{Ry \to \forall x Sx}$ $\quad \dfrac{Ra \to Sx}{Ra \to \forall x Sx}$ $\quad \dfrac{\forall x Rx \to Sx}{\forall x Rx \to \forall x Sx}$ $\quad$ No! $\dfrac{Rx \to Sx}{Rx \to \forall x Sx}$

In the leftmost three cases, for one reason or another, the variable x is not free in the antecedent of the premise. Only in the last case is the variable for which the

quantifier is introduced free in the antecedent of the premise. So the rightmost case violates the constraint, and is not a legitimate application of Gen. Continue to move freely between an expression and its abbreviated forms with justification, abv. That is it!

Because the axioms and rule from before remain available, nothing blocks reasoning with sentential forms as before. Thus, for example, $\forall xRx \to \forall xRx$ and, more generally, $\forall x\mathcal{A} \to \forall x\mathcal{A}$ are of the form $\mathcal{A} \to \mathcal{A}$, and we might derive them by exactly the five steps for T3.1 above. Or, we might just write them down with justification, T3.1. Similarly any theorem from the sentential fragment of *AD* is a theorem of larger quantificational part. Here is a way to get $\forall xRx \to \forall xRx$ without either A1 or A2.

(M) $\vdash_{AD} \forall xRx \to \forall xRx$

1.	$\forall xRx \to Rx$	A4
2.	$\forall xRx \to \forall xRx$	1 Gen

The x is sure to be free for x in Rx. So (1) is an instance of A4. And the only instances of x are bound in $\forall xRx$. So the application of Gen satisfies its constraint. The reasoning is similar in the more general case.

(N) $\vdash_{AD} \forall x\mathcal{A} \to \forall x\mathcal{A}$

1.	$\forall x\mathcal{A} \to \mathcal{A}$	A4
2.	$\forall x\mathcal{A} \to \forall x\mathcal{A}$	1 Gen

Again, $\mathcal{A}^x_x$ is $\mathcal{A}$, and since only free instances of x are "replaced," none of the replacing instances of x is bound in the result. So x is free for x in $\mathcal{A}$, and (1) is therefore a legitimate instance of A4. Because its main operator is an x-quantifier, no instance of x can be free in $\forall x\mathcal{A}$. So we move directly to (2) by Gen.

Here are a few more examples.

T3.27. $\vdash_{AD} \forall x\mathcal{A} \to \forall y\mathcal{A}^x_y$ — where y is not free in $\forall x\mathcal{A}$ but free for x in $\mathcal{A}$

1.	$\forall x\mathcal{A} \to \mathcal{A}^x_y$	A4
2.	$\forall x\mathcal{A} \to \forall y\mathcal{A}^x_y$	1 Gen

The results of derivations (M) and (N) are instances of this more general principle. The difference is that T3.27 makes room for for variable exchange. Given the constraints, this derivation works for exactly the same reasons as the ones before. If y is free for x in $\mathcal{A}$, then (1) is a straightforward instance of A4. And if y is not free in $\forall x\mathcal{A}$, the constraint on Gen is sure to be met. A simple instance of T3.27 in $\mathcal{L}_q$ is $\vdash_{AD} \forall xRx \to \forall yRy$. If you are confused about restrictions on the axiom and rule, think about the derivation as applied to this case. While our quantified instances

of T3.1 could have been derived by sentential rules, T3.27 cannot; $\forall x\mathcal{A} \to \forall x\mathcal{A}$ has sentential form $\mathcal{A} \to \mathcal{A}$; but when x is not the same as y, $\forall x\mathcal{A} \to \forall y\mathcal{A}^x_y$ has sentential form, $\mathcal{A} \to \mathcal{B}$.

T3.28. $\mathcal{A} \vdash_{AD} \forall x\mathcal{A}$ — a derived Gen*

1.	$\mathcal{A}$	prem
2.	$\mathcal{A} \to ([\forall y(y = y) \to \forall y(y = y)] \to \mathcal{A})$	A1
3.	$[\forall y(y = y) \to \forall y(y = y)] \to \mathcal{A}$	2,1 MP
4.	$[\forall y(y = y) \to \forall y(y = y)] \to \forall x\mathcal{A}$	3 Gen
5.	$\forall y(y = y) \to \forall y(y = y)$	T3.1
6.	$\forall x\mathcal{A}$	4,5 MP

In this derivation, we use $\forall y(y = y) \to \forall y(y = y)$ as a mere "dummy" to bring Gen into play. For step (4), it is important that this expression has no free variables, and so no free occurrences of x. For step (6), it is important that it is a theorem, and so can be asserted on line (5). This theorem is so frequently used that we think of it as a derived form of Gen (Gen*).

*T3.29. $\vdash_{AD} \mathcal{A}^x_t \to \exists x\mathcal{A}$ — for any term t free for x in $\mathcal{A}$

Hint: As in sentential cases, show the unabbreviated form, $\mathcal{A}^x_t \to \sim\forall x\sim\mathcal{A}$ and get the final result by abv. You should find $\forall x\sim\mathcal{A} \to \sim\mathcal{A}^x_t$ to be a useful instance of A4. Notice that $[\sim\mathcal{A}]^x_t$ is the same expression as $\sim[\mathcal{A}^x_t]$, as all the replacements must go on inside the $\mathcal{A}$.

T3.30. $\mathcal{A} \to \mathcal{B} \vdash_{AD} \exists x\mathcal{A} \to \mathcal{B}$ — where x is not free in $\mathcal{B}$.

Hint: Similarly, go for an unabbreviated form, and then get the goal by abv.

T3.31. $\vdash_{AD} \forall x(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \forall x\mathcal{B})$ — where x is not free in $\mathcal{A}$

Hint: consider uses of T3.21 and T3.22.

This completes the fragment of *AD* for sentential operators and quantifiers. It remains to add axioms for equality.

*E3.5. Provide derivations for T3.29, T3.30 and T3.31, explaining in words for every step that has a restriction, how you know that that restriction is met.

E3.6. Provide derivations to show each of the following.

*a. $\forall x(Hx \to Rx), \forall yHy \vdash_{AD} \forall zRz$

b. $\forall y(Fy \to Gy) \vdash_{AD} \exists zFz \to \exists xGx$

*c. $\vdash_{AD} \exists x\forall yRxy \to \forall y\exists xRxy$

d. $\forall y\forall x(Fx \to By) \vdash_{AD} \forall y(\exists xFx \to By)$

e. $\vdash_{AD} \exists x(Fx \to \forall yGy) \to \exists x\forall y(Fx \to Gy)$

3.3.2 Equality

We complete our axiomatic derivation system *AD* with three axioms governing equality. In this case, the axioms assert particularly simple, or basic, facts. For any variables $\mathscr{x}_1 \ldots \mathscr{x}_n$ and $\mathscr{y}$, n-place function symbol $\mathscr{h}^n$ and n-place relation symbol $\mathscr{R}^n$, the following forms are axioms.

AE A5. $(\mathscr{y} = \mathscr{y})$

A6. $(\mathscr{x}_i = \mathscr{y}) \to (\mathscr{h}^n\mathscr{x}_1 \ldots \mathscr{x}_i \ldots \mathscr{x}_n = \mathscr{h}^n\mathscr{x}_1 \ldots \mathscr{y} \ldots \mathscr{x}_n)$

A7. $(\mathscr{x}_i = \mathscr{y}) \to (\mathscr{R}^n\mathscr{x}_1 \ldots \mathscr{x}_i \ldots \mathscr{x}_n \to \mathscr{R}^n\mathscr{x}_1 \ldots \mathscr{y} \ldots \mathscr{x}_n)$

From A5, $(x = x)$ and $(z = z)$ are axioms. Of course, these are abbreviations for $=xx$ and $=zz$. This should be straightforward. The others are complicated only by abstract presentation. For A6, $\mathscr{h}^n\mathscr{x}_1 \ldots \mathscr{x}_i \ldots \mathscr{x}_n$ differs from $\mathscr{h}^n\mathscr{x}_1 \ldots \mathscr{y} \ldots \mathscr{x}_n$ just in that variable $\mathscr{x}_i$ is replaced by variable $\mathscr{y}$. $\mathscr{x}_i$ may be any of the variables in $\mathscr{x}_1 \ldots \mathscr{x}_n$. Thus, for example,

(O) $\quad (x = y) \to (f^1x = f^1y) \qquad (x = y) \to (f^3wxy = f^3wyy)$

are simple examples of A6. In the one case, we have a "string" of one variables and replace the only member based on the equality. In the other case, the string is of three variables, and we replace the second. Similarly, $\mathscr{R}^n\mathscr{x}_1 \ldots \mathscr{x}_i \ldots \mathscr{x}_n$ differs from $\mathscr{R}^n\mathscr{x}_1 \ldots \mathscr{y} \ldots \mathscr{x}_n$ just in that variable $\mathscr{x}_i$ is replaced by $\mathscr{y}$. $\mathscr{x}_i$ may be any of the variables in $\mathscr{x}_1 \ldots \mathscr{x}_n$. Thus, for example,

(P) $\quad (x = z) \to (A^1x \to A^1z) \qquad (z = w) \to (A^2xz \to A^2xw)$

are simple examples of A7.

This completes the axioms and rules of our full derivation system *AD*. As examples, let us begin with some fundamental principles of equality. Suppose that $\mathscr{r}$, $\mathscr{s}$ and $\mathscr{t}$ are arbitrary terms.

T3.32. $\vdash_{AD} (t = t)$ — *reflexivity* of equality

1.	$y = y$	A5
2.	$\forall y(y = y)$	1 Gen*
3.	$\forall y(y = y) \rightarrow (t = t)$	A4
4.	$t = t$	3,2 MP

Since $y = y$ has no quantifiers, any term t is sure to be free for y in it. So (3) is sure to be an instance of A4. This theorem strengthens A5 insofar as the axiom applies only to variables, but the theorem has application to arbitrary terms. Thus $(z = z)$ is an instance of the axiom, but $(f^2xy = f^2xy)$ is an instance of the theorem as well. We *convert* variables to terms by Gen* with A4 and MP. This pattern repeats in the following.

T3.33. $\vdash_{AD} (t = s) \rightarrow (s = t)$ — *symmetry* of equality

1.	$(x = y) \rightarrow [(x = x) \rightarrow (y = x)]$	A7
2.	$(x = x)$	A5
3.	$(x = y) \rightarrow (y = x)$	1,2 T3.6
4.	$\forall x[(x = y) \rightarrow (y = x)]$	3 Gen*
5.	$\forall x[(x = y) \rightarrow (y = x)] \rightarrow [(t = y) \rightarrow (y = t)]$	A4
6.	$(t = y) \rightarrow (y = t)$	5,4 MP
7.	$\forall y[(t = y) \rightarrow (y = t)]$	6 Gen*
8.	$\forall y[(t = y) \rightarrow (y = t)] \rightarrow [(t = s) \rightarrow (s = t)]$	A4
9.	$(t = s) \rightarrow (s = t)$	8,7 MP

In (1), $x = x$ is (an abbreviation of an expression) of the form $\mathscr{R}^2xx$, and $y = x$ is of that same form with the first instance of x replaced by y. Thus (1) is an instance of A7. At line (3) we have symmetry expressed at the level of variables. Then the task is just to convert from variables to terms as before. Notice that, again, (5) and (8) are legitimate applications of A4 insofar as there are no quantifiers in the consequents.

T3.34. $\vdash_{AD} (r = s) \rightarrow [(s = t) \rightarrow (r = t)]$ — *transitivity* of equality

Hint: Start with $(y = x) \rightarrow [(y = z) \rightarrow (x = z)]$ as an instance of A7 — being sure that you see how it *is* an instance of A7. Then you can use T3.33 to get $(x = y) \rightarrow [(y = z) \rightarrow (x = z)]$, and all you have to do is convert from variables to terms as above.

T3.35. $r = s, s = t \vdash_{AD} r = t$

Hint: This is a mere recasting of T3.34 and follows directly from it.

T3.36. $\vdash_{AD} (t_i = s) \rightarrow (h^n t_1 \ldots t_i \ldots t_n = h^n t_1 \ldots s \ldots t_n)$

Hint: For any given instance of this theorem, you can start with $(x_i = y) \rightarrow (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$ as an instance of A6. Then it is easy to convert $x_1 \ldots x_n$ to $t_1 \ldots t_n$, and y to s.

T3.37. $\vdash_{AD} (t_i = s) \rightarrow (\mathcal{R}^n t_1 \ldots t_i \ldots t_n \rightarrow \mathcal{R}^n t_1 \ldots s \ldots t_n)$

Hint: As for T3.36, for any given instance of this theorem, you can start with $(x_i = y) \rightarrow (\mathcal{R}^n x_1 \ldots x_i \ldots x_n \rightarrow \mathcal{R}^n x_1 \ldots y \ldots x_n)$ as an instance of A7. Then it is easy to convert $x_1 \ldots x_n$ to $t_1 \ldots t_n$, and y to s.

We will see further examples in the context of the extended application to come in the next section.

E3.7. Provide demonstrations for T3.34 and T3.35.

E3.8. Provide demonstrations for the following instances of T3.36 and T3.37. Then, in each case, say in words how you would go about showing the results for an arbitrary number of places.

a. $(f^1 x = g^2 xy) \rightarrow (h^3 z f^1 x f^1 z = h^3 z g^2 xy f^1 z)$

*b. $(s = t) \rightarrow (\mathcal{A}^2 r s \rightarrow \mathcal{A}^2 r t)$

3.3.3 Peano Arithmetic

$\mathcal{L}_{\mathrm{NT}}$ is a language like $\mathcal{L}_{\mathrm{NT}}^{<}$ introduced from section 2.2.5 on p. 63 but without the $<$ symbol: There are the constant symbol $\emptyset$, the function symbols S, $+$ and $\times$, and the relation symbol $=$. It is possible to treat $x \leq y$ as an abbreviation for $\exists v(v + x = y)$ and $x < y$ as an abbreviation for $\exists v(Sv + x) = y$ (these definitions are summarized in the language of arithmetic reference, p. 303). Officially, formulas of this language are so far uninterpreted. It is natural, however, to think of them with their usual meanings, with $\emptyset$ for zero, S the successor function, $+$ the addition function, $\times$ the multiplication function, and $=$ the equality relation. But, again, we do not need to think about that for now.

AD Quick Reference

AD A1. $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$

A2. $(\mathcal{O} \to (\mathcal{P} \to \mathcal{Q})) \to ((\mathcal{O} \to \mathcal{P}) \to (\mathcal{O} \to \mathcal{Q}))$

A3. $(\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$

A4. $\forall x\mathcal{P} \to \mathcal{P}^x_t$ — where t is free for x in $\mathcal{P}$

A5. $(x = x)$

A6. $(x_i = y) \to (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$

A7. $(x_i = y) \to (\mathcal{R}^n x_1 \ldots x_i \ldots x_n \to \mathcal{R}^n x_1 \ldots y \ldots x_n)$

MP $\mathcal{P} \to \mathcal{Q}, \mathcal{P} \vdash \mathcal{Q}$

Gen $\mathcal{A} \to \mathcal{B} \vdash \mathcal{A} \to \forall x\mathcal{B}$ — where x is not free in $\mathcal{A}$

T3.1 $\vdash_{AD} \mathcal{A} \to \mathcal{A}$

T3.2 $\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C} \vdash_{AD} \mathcal{A} \to \mathcal{C}$

T3.3 $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{AD} \mathcal{B} \to (\mathcal{A} \to \mathcal{C})$

T3.4 $\vdash_{AD} (\mathcal{B} \to \mathcal{C}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$

T3.5 $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to [(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to \mathcal{C})]$

T3.6 $\mathcal{B}, \mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{AD} \mathcal{A} \to \mathcal{C}$

T3.7 $\vdash_{AD} (\sim\mathcal{A} \to \mathcal{A}) \to \mathcal{A}$

T3.8 $\vdash_{AD} (\sim\mathcal{B} \to \sim\mathcal{A}) \to (\mathcal{A} \to \mathcal{B})$

T3.9 $\vdash_{AD} \sim\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$

T3.10 $\vdash_{AD} \sim\sim\mathcal{A} \to \mathcal{A}$

T3.11 $\vdash_{AD} \mathcal{A} \to \sim\sim\mathcal{A}$

T3.12 $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to (\sim\sim\mathcal{A} \to \sim\sim\mathcal{B})$

T3.13 $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to (\sim\mathcal{B} \to \sim\mathcal{A})$

T3.14 $\vdash_{AD} (\sim\mathcal{A} \to \mathcal{B}) \to (\sim\mathcal{B} \to \mathcal{A})$

T3.15 $\vdash_{AD} (\mathcal{A} \to \mathcal{B}) \to [(\sim\mathcal{A} \to \mathcal{B}) \to \mathcal{B}]$

T3.16 $\vdash_{AD} \mathcal{A} \to [\sim\mathcal{B} \to \sim(\mathcal{A} \to \mathcal{B})]$

T3.17 $\vdash_{AD} \mathcal{A} \to (\mathcal{A} \vee \mathcal{B})$

T3.18 $\vdash_{AD} \mathcal{A} \to (\mathcal{B} \vee \mathcal{A})$

T3.19 $\vdash_{AD} (\mathcal{A} \wedge \mathcal{B}) \to \mathcal{B}$

T3.20 $\vdash_{AD} (\mathcal{A} \wedge \mathcal{B}) \to \mathcal{A}$

T3.21 $\mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{AD} (\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C}$

T3.22 $(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C} \vdash_{AD} \mathcal{A} \to (\mathcal{B} \to \mathcal{C})$

T3.23 $\mathcal{A}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \mathcal{B}$

T3.24 $\mathcal{B}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \mathcal{A}$

T3.25 $\sim\mathcal{A}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \sim\mathcal{B}$

T3.26 $\sim\mathcal{B}, \mathcal{A} \leftrightarrow \mathcal{B} \vdash_{AD} \sim\mathcal{A}$

T3.27 $\vdash_{AD} \forall x\mathcal{A} \to \forall y\mathcal{A}^x_y$

where y is not free in $\forall x\mathcal{A}$ but is free for x in $\mathcal{A}$

T3.28 $\mathcal{A} \vdash_{AD} \forall x\mathcal{A}$ (Gen*)

T3.29 $\vdash_{AD} \mathcal{A}^x_t \to \exists x\mathcal{A}$

where t is free for x in $\mathcal{A}$

T3.30 $\mathcal{A} \to \mathcal{B} \vdash_{AD} \exists x\mathcal{A} \to \mathcal{B}$

where x is not free in $\mathcal{B}$

T3.31 $\vdash_{AD} \forall x(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \forall x\mathcal{B})$

where x is not free in $\mathcal{A}$

T3.32 $\vdash_{AD} (t = t)$

T3.33 $\vdash_{AD} (t = s) \to (s = t)$

T3.34 $\vdash_{AD} (r = s) \to [(s = t) \to (r = t)]$

T3.35 $r = s,\ s = t \vdash_{AD} r = t$

T3.36 $\vdash_{AD} (t_i = s) \to (h^n t_1 \ldots t_i \ldots t_n = h^n t_1 \ldots s \ldots t_n)$

T3.37 $\vdash_{AD} (t_i = s) \to (\mathcal{R}^n t_1 \ldots t_i \ldots t_n \to \mathcal{R}^n t_1 \ldots s \ldots t_n)$

We will say that a formula $\mathcal{P}$ is an *AD theorem of Peano Arithmetic* just in case $\mathcal{P}$ follows in *AD* given as premises the following axioms for Peano Arithmetic.[2] These axioms are presented as formulas with free variables; but with Gen* and A4, they are equivalent to universally quantified forms — and we might as well have stated the axioms as universally quantified sentences.

PA
1. $\sim(Sx = \emptyset)$
2. $(Sx = Sy) \rightarrow (x = y)$
3. $(x + \emptyset) = x$
4. $(x + Sy) = S(x + y)$
5. $(x \times \emptyset) = \emptyset$
6. $(x \times Sy) = [(x \times y) + x]$
7. $[\mathcal{P}^{x}_{\emptyset} \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^{x}_{Sx})] \rightarrow \forall x\mathcal{P}$

In the ordinary case we suppress mention of PA1 - PA7 as premises, and simply write PA $\vdash_{AD} \mathcal{P}$ to indicate that $\mathcal{P}$ is an *AD* theorem of Peano arithmetic — that there is an *AD* derivation of $\mathcal{P}$ which may include appeal to any of PA1 - PA7.

The axioms set up basic arithmetic on the non-negative integers. Intuitively, $\emptyset$ is not the successor of any non-negative integer (PA1); if the successor of x is the same as the successor of y, then x is y (PA2); x plus $\emptyset$ is equal to x (PA3); x plus one more than y is equal to one more than x plus y (PA4); x times $\emptyset$ is equal to $\emptyset$ (PA5); x times one more than y is equal to x times y plus x (PA6); and if $\mathcal{P}$ applies to $\emptyset$, and for any x, if $\mathcal{P}$ applies to x, then it also applies to Sx, then $\mathcal{P}$ applies to every x (PA7). This last form represents the *principle of mathematical induction*. Strictly, it is an *axiom schema* insofar as indefinitely many formulas might be of that form.

Sometimes it is convenient to have the principle of mathematical induction in rule form.

T3.38. $\mathcal{P}^{x}_{\emptyset}, \forall x(\mathcal{P} \rightarrow \mathcal{P}^{x}_{Sx})$, PA $\vdash_{AD} \forall x\mathcal{P}$ — (a derived Ind*)

1.	$\mathcal{P}^{x}_{\emptyset}$	prem
2.	$\forall x(\mathcal{P} \rightarrow \mathcal{P}^{x}_{Sx})$	prem
3.	$[\mathcal{P}^{x}_{\emptyset} \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^{x}_{Sx})] \rightarrow \forall x\mathcal{P}$	PA7
4.	$\mathcal{P}^{x}_{\emptyset} \rightarrow [\forall x(\mathcal{P} \rightarrow \mathcal{P}^{x}_{Sx}) \rightarrow \forall x\mathcal{P}]$	3 T3.22
5.	$\forall x(\mathcal{P} \rightarrow \mathcal{P}^{x}_{Sx}) \rightarrow \forall x\mathcal{P}$	4,1 MP
6.	$\forall x\mathcal{P}$	5,2 MP

[2] After the work of R. Dedekind and G. Peano. For historical discussion, see Wang, "The Axiomatization of Arithmetic. Observe that 'theorem' is therefore context-relative. A theorem of Peano arithmetic which results only givien PA1 - PA7 is not a theorem of AD just because it takes some of PA1 - PA7 for its derivation.

Observe the way we simply appeal to PA7 as a premise at (3). Again, that we can do this in a derivation, is a consequence of our taking all the axioms available as premises. So if we were to encounter $\mathcal{P}^x_\emptyset$, and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$ in a derivation with the axioms of PA, we could safely move to the conclusion that $\forall x\mathcal{P}$ by this derived rule Ind*. We will have much more to say about the principle of mathematical induction in Part II. For now, it is enough to *recognize* its instances. Thus, for example, if $\mathcal{P}$ is $\sim(x = Sx)$, the corresponding instance of PA7 would be,

(Q) $\quad [\sim(\emptyset = S\emptyset) \wedge \forall x(\sim(x = Sx) \to \sim(Sx = SSx))] \to \forall x{\sim}(x = Sx)$

There is the formula with $\emptyset$ substituted for x, the formula itself, and the formula with Sx substituted for x. If the entire antecedent is satisfied, then the formula holds for *every* x. For the corresponding application of T3.38 you would need $\sim(\emptyset = S\emptyset)$ and $\forall x[\sim(x = Sx) \to \sim(Sx = SSx)]$ in order to move to the conclusion that $\forall x{\sim}(x = Sx)$. You should track these examples through. The principle of mathematical induction turns out to be essential for deriving many general results.

As before, if a theorem is derived from some premises, we use the theorem in derivations that follow. Thus we build toward increasingly complex results. Let us start with some simple generalizations of the premises for application to arbitrary terms. The derivations all follow the Gen* / A4 / MP pattern we have seen before.

T3.39. PA $\vdash_{AD} \sim(St = \emptyset)$

1.	$\sim(Sx = \emptyset)$	PA1
2.	$\forall x{\sim}(Sx = \emptyset)$	1 Gen*
3.	$\forall x{\sim}(Sx = \emptyset) \to \sim(St = \emptyset)$	A4
4.	$\sim(St = \emptyset)$	3,2 MP

As usual, because there is no quantifier in the consequent, (3) is sure to satisfy the constraint on A4, no matter what t may be.

*T3.40. PA $\vdash_{AD} (St = Ss) \to (t = s)$

T3.41. PA $\vdash_{AD} (t + \emptyset) = t$

T3.42. PA $\vdash_{AD} (t + Ss) = S(t + s)$

T3.43. PA $\vdash_{AD} (t \times \emptyset) = \emptyset$

T3.44. PA $\vdash_{AD} (t \times S s) = [(t \times s) + t]$

If a theorem T3.n is an equality $(t = s)$, let T3.n* be $(s = t)$. Thus T3.41* is PA $\vdash_{AD}$ $t = (t + \emptyset)$; T3.42* is PA $\vdash_{AD}$ $S(t + s) = (t + Ss)$. In each case, the result is immediate from the theorem with T3.33 and MP. Notice that t and s in these theorems may be *any* terms. Thus,

(R) $\quad (x + \emptyset) = x \qquad ((x \times y) + \emptyset) = (x \times y) \qquad [(\emptyset + x) + \emptyset] = (\emptyset + x)$

are all straightforward instances of T3.41.

Given this much, we are ready for a series of results which are much more interesting — for example, some general principles of commutativity and associativity. For a first application of Ind*, let $\mathcal{P}$ be $[(\emptyset + x) = x]$; then $\mathcal{P}^x_\emptyset$ is $[(\emptyset + \emptyset) = \emptyset]$ and $\mathcal{P}^x_{Sx}$ is $[(\emptyset + Sx) = Sx]$.

T3.45. PA $\vdash_{AD} (\emptyset + t) = t$

1.	$(\emptyset + \emptyset) = \emptyset$	T3.41
2.	$[(\emptyset + x) = x] \to [S(\emptyset + x) = Sx]$	T3.36
3.	$[S(\emptyset + x) = (\emptyset + Sx)]$	T3.42*
4.	$[S(\emptyset + x) = (\emptyset + Sx)] \to [(S(\emptyset + x) = Sx) \to ((\emptyset + Sx) = Sx)]$	T3.37
5.	$(S(\emptyset + x) = Sx) \to ((\emptyset + Sx) = Sx)$	4,3 MP
6.	$[(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx]$	2,5 T3.2
7.	$\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$	6 Gen*
8.	$\forall x[(\emptyset + x) = x]$	1,7 Ind*
9.	$\forall x[(\emptyset + x) = x] \to [(\emptyset + t) = t]$	A4
10.	$[(\emptyset + t) = t]$	9,8 MP

The key to this derivation, and others like it, is bringing Ind* into play. The basic strategy for the beginning and end of these arguments is always the same. In this case,

(S)

1.	$(\emptyset + \emptyset) = \emptyset$	T3.41
$\vdots$		
6.	$[(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx]$	
7.	$\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$	6 Gen*
8.	$\forall x[(\emptyset + x) = x]$	1,7 Ind*
9.	$\forall x[(\emptyset + x) = x] \to [(\emptyset + t) = t]$	A4
10.	$[(\emptyset + t) = t]$	9,8 MP

The goal is automatic by A4 and MP once you have $\forall x[(\emptyset + x) = x]$ by Ind* at (8). For this, you need $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$. We have $\mathcal{P}^x_\emptyset$ at (1) as an instance of

T3.41 — and $\mathcal{P}^x_\emptyset$ is almost always easy to get. $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$ is automatic by Gen* from (6). So the real work is getting (6). Thus, once you see what is going on, the entire derivation for T3.45 boils down to lines (2) - (6). For this, begin by noticing that the antecedent of what we want is like the antecedent of (2), and the consequent like what we want but for the equivalence in (3). Given this, it is a simple matter to apply T3.37 to switch the one term for the equivalent one we want.

T3.46. PA $\vdash_{AD}$ $[(St + \emptyset) = S(t + \emptyset)]$

1.	$(St + \emptyset) = St$	T3.41
2.	$t = (t + \emptyset)$	T3.41*
3.	$[t = (t + \emptyset)] \to [St = S(t + \emptyset)]$	T3.36
4.	$St = S(t + \emptyset)$	3,2 MP
5.	$(St + \emptyset) = S(t + \emptyset)$	1,4 T3.35

This derivation has T3.41 at (1) with St for t. Line (2) is a straightforward version of T3.41*. Then the key to the derivation is that the antecedent of (1) is like what we want, and the consequent of (1) is like what we want but for the equality on (2). The goal then is to use T3.36 to switch the one term for the equivalent one. You should get used to this pattern of using T3.36 and T3.37 to substitute terms. This result forms the "zero-case" for the one that follows.

T3.47. PA $\vdash_{AD}$ $[(St + s) = S(t + s)]$

1.	$[(St + \emptyset) = S(t + \emptyset)]$	T3.46
2.	$[(St + x) = S(t + x)] \to [S(St + x) = SS(t + x)]$	T3.36
3.	$[S(St + x) = (St + Sx)]$	T3.42*
4.	$[S(St + x) = (St + Sx)] \to$ $([S(St + x) = SS(t + x)] \to [(St + Sx) = SS(t + x)])$	T3.37
5.	$[S(St + x) = SS(t + x)] \to [(St + Sx) = SS(t + x)]$	4,3 MP
6.	$[(St + x) = S(t + x)] \to [(St + Sx) = SS(t + x)]$	2,5 T3.2
7.	$[S(t + x) = (t + Sx)]$	T3.42*
8.	$[S(t + x) = (t + Sx)] \to [SS(t + x) = S(t + Sx)]$	T3.36
9.	$[SS(t + x) = S(t + Sx)]$	8,7 MP
10.	$[SS(t + x) = S(t + Sx)] \to$ $([(St + Sx) = SS(t + x)] \to [(St + Sx) = S(t + Sx)])$	T3.37
11.	$[(St + Sx) = SS(t + x)] \to [(St + Sx) = S(t + Sx)]$	10,9 MP
12.	$[(St + x) = S(t + x)] \to [(St + Sx) = S(t + Sx)]$	6,11 T3.2
13.	$\forall x([(St + x) = S(t + x)] \to [(St + Sx) = S(t + Sx)])$	12 Gen*
14.	$\forall x[(St + x) = S(t + x)]$	1,13 Ind*
15.	$\forall x[(St + x) = S(t + x)] \to [(St + s) = S(t + s)]$	A4
16.	$[(St + s) = S(t + s)]$	15,14 MP

The idea behind this longish derivation is to bring Ind* into play, where formula $\mathcal{P}$ is, $[(St + x) = S(t + x)]$. Do not worry about how we got this for now. Given this much, the following setup is automatic,

(T)

1.	$[(St + \emptyset) = S(t + \emptyset)]$	T3.46
	$\vdots$	
12.	$[(St + x) = S(t + x)] \rightarrow [(St + Sx) = S(t + Sx)]$	
13.	$\forall x([(St + x) = S(t + x)] \rightarrow [(St + Sx) = S(t + Sx)])$	12 Gen*
14.	$\forall x[(St + x) = S(t + x)]$	1,13 Ind*
15.	$\forall x[(St + x) = S(t + x)] \rightarrow [(St + s) = S(t + s)]$	A4
16.	$[(St + s) = S(t + s)]$	15,14 MP

We have the zero-case from T3.46 on (1); the goal is automatic once we have the result on (12). For (12), the antecedent at (2) is what we want, and the consequent is right but for the equivalences on (3) and (9). We use T3.37 to substitute terms into the consequent. The equivalence on (3) is a straightforward instance of T3.42*. We had to work (just a bit) starting again with T3.42* to get the equivalence on (9).

T3.48. PA $\vdash_{AD}$ $[(t + s) = (s + t)]$ — *commutativity of addition*

1.	$[(t + \emptyset) = t]$	T3.41
2.	$[t = (\emptyset + t)]$	T3.45*
3.	$[(t + \emptyset) = (\emptyset + t)]$	1,2 T3.35
4.	$[(t + x) = (x + t)] \rightarrow [S(t + x) = S(x + t)]$	T3.36
5.	$[S(t + x) = (t + Sx)]$	T3.42*
6.	$[S(t + x) = (t + Sx)] \rightarrow$ ($[S(t + x) = S(x + t)] \rightarrow [(t + Sx) = S(x + t)]$)	T3.37
7.	$[S(t + x) = S(x + t)] \rightarrow [(t + Sx) = S(x + t)]$	6,5 MP
8.	$[(t + x) = (x + t)] \rightarrow [(t + Sx) = S(x + t)]$	4,7 T3.2
9.	$[S(x + t) = (Sx + t)]$	T3.47*
10.	$[S(x + t) = (Sx + t)] \rightarrow$ ($[(t + Sx) = S(x + t)] \rightarrow [(t + Sx) = (Sx + t)]$)	T3.37
11.	$[(t + Sx) = S(x + t)] \rightarrow [(t + Sx) = (Sx + t)]$	10,9 MP
12.	$[(t + x) = (x + t)] \rightarrow [(t + Sx) = (Sx + t)]$	8,11 T3.2
13.	$\forall x([(t + x) = (x + t)] \rightarrow [(t + Sx) = (Sx + t)])$	12 Gen*
14.	$\forall x[(t + x) = (x + t)]$	3,13 Ind*
15.	$\forall x[(t + x) = (x + t)] \rightarrow [(t + s) = (s + t)]$	A4
16.	$[(t + s) = (s + t)]$	15,14 MP

The pattern of this derivation is very much like ones we have seen before. Where $\mathcal{P}$ is $[(t + x) = (x + t)]$ we have the zero-case at (3), and the derivation effectively reduces to getting (12). We get this by substituting into the consequent of (4) by means of the equivalences on (5) and (9).

T3.49. PA $\vdash_{AD}$ $[((r + s) + \emptyset) = (r + (s + \emptyset))]$

Hint: Begin with $[((r + s) + \emptyset) = (r + s)]$ as an instance of T3.41. The derivation is then a matter of using T3.41* to replace s in the right-hand side with $(s + \emptyset)$.

*T3.50. PA $\vdash_{AD}$ $[((r + s) + t) = (r + (s + t))]$ — *associativity of addition*

Hint: For an application of Ind*, let $\mathscr{P}$ be $[((r + s) + x) = (r + (s + x))]$. Start with $[((r+s)+x) = (r+(s+x))] \rightarrow [S((r+s)+x) = S(r+(s+x))]$ as an instance of T3.36, and substitute into the consequent as necessary by T3.42* to reach $[((r + s) + x) = (r + (s + x))] \rightarrow [((r + s) + Sx)) = (r + (s + Sx))]$. The derivation is longish, but straightforward.

T3.51. PA $\vdash_{AD}$ $[(\emptyset \times t) = \emptyset]$

Hint: For an application of Ind*, let $\mathscr{P}$ be $[(\emptyset \times x) = \emptyset]$; then the derivation reduces to sowing $[(\emptyset \times x) = \emptyset] \rightarrow [(\emptyset \times Sx) = \emptyset]$. This is easy enough if you use T3.41* and T3.44* to show that $[(\emptyset \times x) = (\emptyset \times Sx)]$.

T3.52. PA $\vdash_{AD}$ $[(St \times \emptyset) = ((t \times \emptyset) + \emptyset)]$

Hint: This does not require application of Ind*.

*T3.53. PA $\vdash_{AD}$ $[(St \times s) = ((t \times s) + s)]$

Hint: For an application of Ind*, let $\mathscr{P}$ be $[(St \times x) = ((t \times x) + x)]$. The derivation reduces to getting $[(St \times x) = ((t \times x) + x)] \rightarrow [(St \times Sx) = ((t \times Sx) + Sx)]$. For this, you can start with $[(St \times x) = ((t \times x) + x)] \rightarrow [((St\times x)+St) = (((t\times x)+x)+St)]$ as an instance of T3.36, and substitute into the consequent. You may find it helpful to obtain $[(x + St) = (t + Sx)]$ and then $[((t\times x)+(x + St)) = ((t\times Sx)+Sx)]$ as as a preliminary result.

T3.54. PA $\vdash_{AD}$ $[(t \times s) = (s \times t)]$ — *commutativity of multiplication*

Hint: For an application of Ind*, let $\mathscr{P}$ be $[(t \times x) = (x \times t)]$. You can start with $[(t \times x) = (x \times t)] \rightarrow [((t \times x) + t) = ((x \times t) + t)]$ as an instance of T3.36, and substitute into the consequent.

We will stop here. With the derivation system *ND* of chapter 6, we obtain all these results and more. But that system is easier to manipulate than what we have so far in *AD*. Still, we have obtained some significant results! Perhaps you have heard from your mother's knee that $a + b = b + a$. But this is a sweeping general claim of the sort that cannot ever have all its instances checked. We have derived it from the Peano axioms. Of course, one might want to know about justifications for the Peano axioms. But that is another story.

*E3.9. Provide derivations to show each of T3.40, T3.41, T3.42, T3.43, T3.44, T3.49, T3.50, T3.51, T3.52, T3.53, and T3.54. Hint: you may find the AD Peano reference on p. 95 helpful.

E3.10. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. A *consequence* in a some axiomatic logic of Γ, and then a consequence in *AD* of Γ.

b. An *AD* theorem of Peano arithmetic.

c. Term t being *free for* variable x in formula $\mathcal{A}$ along with the restrictions on A4 and Gen.

Peano Arithmetic (AD)

PA
1. $\sim(Sx = \emptyset)$
2. $(Sx = Sy) \rightarrow (x = y)$
3. $(x + \emptyset) = x$
4. $(x + Sy) = S(x + y)$
5. $(x \times \emptyset) = \emptyset$
6. $(x \times Sy) = [(x \times y) + x]$
7. $[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})] \rightarrow \forall x\mathcal{P}$

T3.38 $\mathcal{P}^x_\emptyset, \forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$, PA $\vdash_{AD} \forall x\mathcal{P}$ — Ind*

T3.39 PA $\vdash_{AD} \sim(St = \emptyset)$

T3.40 PA $\vdash_{AD} (St = Ss) \rightarrow (t = s)$

T3.41 PA $\vdash_{AD} (t + \emptyset) = t$

T3.42 PA $\vdash_{AD} (t + Ss) = S(t + s)$

T3.43 PA $\vdash_{AD} (t \times \emptyset) = \emptyset$

T3.44 PA $\vdash_{AD} (t \times Ss) = [(t \times s) + t]$

T3.45 PA $\vdash_{AD} (\emptyset + t) = t$

T3.46 PA $\vdash_{AD} [(St + \emptyset) = S(t + \emptyset)]$

T3.47 PA $\vdash_{AD} [(St + s) = S(t + s)]$

T3.48 PA $\vdash_{AD} [(t + s) = (s + t)]$ — *commutativity of addition*

T3.49 PA $\vdash_{AD} [((r + s) + \emptyset) = (r + (s + \emptyset))]$

T3.50 PA $\vdash_{AD} [((r + s) + t) = (r + (s + t))]$ — *associativity of addition*

T3.51 PA $\vdash_{AD} [(\emptyset \times t) = \emptyset]$

T3.52 PA $\vdash_{AD} [(St \times \emptyset) = ((t \times \emptyset) + \emptyset)]$

T3.53 PA $\vdash_{AD} [(St \times s) = ((t \times s) + s)]$

T3.54 PA $\vdash_{AD} [(t \times s) = (s \times t)]$ — *commutativity of multiplication*

If T3.*n* is of the sort $(t = s)$, then T3.*n** is $(s = t)$.

Chapter 4

Semantics

Having introduced the grammar for our formal languages and even (if you did not skip the last chapter) done derivations in them, we need to say something about *semantics* — about the conditions under which their expressions are true and false. In addition to *logical validity* from chapter 1 and *validity in AD* from chapter 3, this will lead to a third, *semantic* notion of validity. Again, the discussion divides into the relatively simple sentential case, and then the full quantificational version. Recall that we are introducing formal languages in their "pure" form, apart from associations with ordinary language. Having discussed, in this chapter, conditions under which formal expressions are true and not, in the next chapter, we will finally turn to translation, and so to ways formal expressions are associated with ordinary ones.

4.1 Sentential

Let us say that any sentence in a sentential or quantificational language with no subformula (other than itself) that is a sentence is *basic*. For a sentential language, basic sentences are the sentence letters, for a sentence letter is precisely a sentence with no subformula other than itself that is a sentence. In the quantificational case, basic sentences may be more complex.[1] In this part, we treat basic sentences as atomic. Our initial focus is on forms with just operators $\sim$ and $\rightarrow$. We begin with an account of the conditions under which sentences are true and not true, learn to apply that account in arbitrary conditions, and turn to validity. The section concludes with applications to our abbreviations, $\wedge$, $\vee$, and $\leftrightarrow$.

[1]Thus the basic sentences of $A \wedge B$ are just the atomic subformulas A and B. But $Fa \wedge \exists xGx$, say has atomic subformulas Fa and Gx, but basic parts Fa and $\exists xGx$.

4.1.1 Interpretations and Truth

Sentences are true and false relative to an *interpretation* of basic sentences. In the sentential case, the notion of an interpretation is particularly simple. For any formal language $\mathscr{L}$, a *sentential interpretation* assigns a truth value *true* or *false*, T or F, to each of its basic sentences. Thus, for $\mathscr{L}_s$ we might have interpretations I and J,

(A)

	A	B	C	D	E	F	G	H	
I	T	T	T	T	T	T	T	T	...
	A	B	C	D	E	F	G	H	
J	T	T	F	F	T	T	F	F	...

When a sentence $\mathscr{A}$ is T on an interpretation I, we write I[$\mathscr{A}$] = T, and when it is F, we write, I[$\mathscr{A}$] = F. Thus, in the above case, J[B] = T and J[C] = F.

Truth for complex sentences depends on truth and falsity for their parts. In particular, for any interpretation I,

ST (∼) For any sentence $\mathscr{P}$, I[$\sim\mathscr{P}$] = T iff I[$\mathscr{P}$] = F; otherwise I[$\sim\mathscr{P}$] = F.

(→) For any sentences $\mathscr{P}$ and $\mathscr{Q}$, I[$(\mathscr{P} \to \mathscr{Q})$] = T iff I[$\mathscr{P}$] = F or I[$\mathscr{Q}$] = T (or both); otherwise I[$(\mathscr{P} \to \mathscr{Q})$] = F.

Thus a basic sentence is true or false depending on the interpretation. For complex sentences, $\sim\mathscr{P}$ is true iff $\mathscr{P}$ is not true; and $(\mathscr{P} \to \mathscr{Q})$ is true iff $\mathscr{P}$ is not true or $\mathscr{Q}$ is. (In the quantificational case, we will introduce a notion of *satisfaction* distinct from truth. However, in the sentential case, satisfaction and truth are the same: An arbitrary sentence $\mathscr{A}$ is *satisfied* on a sentential interpretation I iff it is true on I. So definition ST is all we need.)

It is traditional to represent the information from ST(∼) and ST(→) in the following *truth tables*.

T(∼)

$\mathscr{P}$	$\sim\mathscr{P}$
T	***F***
F	***T***

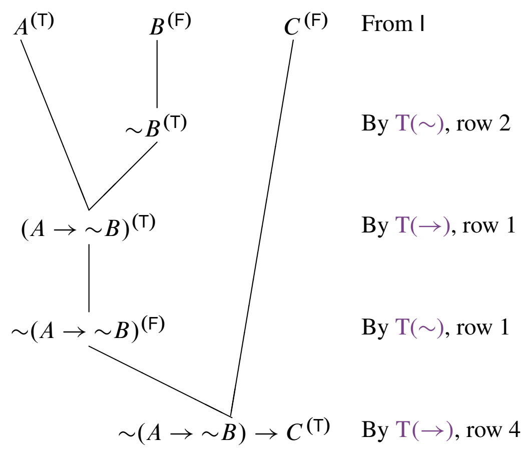
T(→)

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \to \mathscr{Q}$
T	T	***T***
T	F	***F***
F	T	***T***
F	F	***T***

From ST(∼), we have that if $\mathscr{P}$ is F then $\sim\mathscr{P}$ is T; and if $\mathscr{P}$ is T then $\sim\mathscr{P}$ is F. This is just the way to read table T(∼) from left-to-right in the bottom row, and then the top row. Similarly, from ST(→), we have that $\mathscr{P} \to \mathscr{Q}$ is T in conditions represented by the first, third and fourth rows of T(→). The only way for $\mathscr{P} \to \mathscr{Q}$ to be F is when $\mathscr{P}$ is T and $\mathscr{Q}$ is F as in the second row.

ST works recursively. Whether a basic sentence is true comes directly from the interpretation; truth for other sentences depends on truth for their immediate subformulas — and can be read directly off the tables. As usual, we can use trees to see how it works. Suppose $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] = \mathsf{F}$, and $\mathsf{I}[C] = \mathsf{F}$. Then $\mathsf{I}[\sim(A \to \sim B) \to C] = \mathsf{T}$.

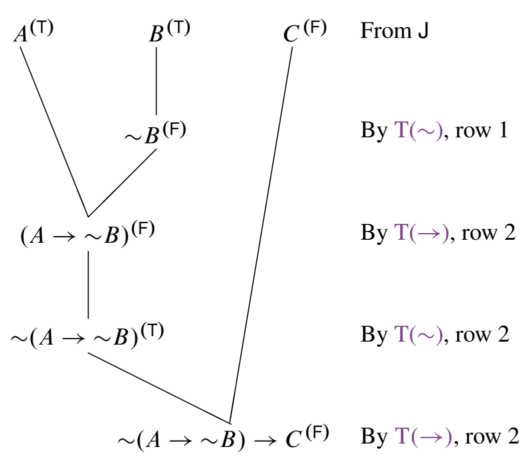
(B)



The basic tree is the same as to show that $\sim(A \to \sim B) \to C$ is a formula. From the interpretation, A is T, B is F, and C is F. These are across the top. Since B is F, from the bottom row of table T(∼), $\sim B$ is T. Since A is T and $\sim B$ is T, reading across the top row of the table T(→), $A \to \sim B$ is T. And similarly, according to the tree, for the rest. You should carefully follow each step. As we built the formula from its parts to the whole, so now we calculate its truth from the parts to the whole.

Here is the same formula considered on another interpretation. Where interpretation J is as on p. 97, $\mathsf{J}[\sim(A \to \sim B) \to C] = \mathsf{F}$.

(C)



This time, for both applications of ST($\rightarrow$), the antecedent is T and the consequent is F; thus we are working on the second row of table T($\rightarrow$), and the conditionals evaluate to F. Again, you should follow each step in the tree.

E4.1. Where the interpretation is as J from p. 97, with $\mathsf{J}[A] = \mathsf{T}$, $\mathsf{J}[B] = \mathsf{T}$ and $\mathsf{J}[C] = \mathsf{F}$, use trees to decide whether the following sentences of $\mathcal{L}_s$ are T or F.

*a. $\sim A$	b. $\sim\sim C$
c. $A \rightarrow C$	d. $C \rightarrow A$
*e. $\sim(A \rightarrow A)$	*f. $(\sim A \rightarrow A)$
g. $\sim(A \rightarrow \sim C) \rightarrow C$	h. $(\sim A \rightarrow C) \rightarrow C$
*i. $(A \rightarrow \sim B) \rightarrow \sim(B \rightarrow \sim A)$	j. $\sim(B \rightarrow \sim A) \rightarrow (A \rightarrow \sim B)$

4.1.2 Arbitrary Interpretations

Sentences are true and false relative to an interpretation. But whether an argument is *semantically valid* depends on truth and falsity relative to *every* interpretation. As a first step toward determining semantic validity, in this section, we generalize the method of the last section to calculate truth values relative to arbitrary interpretations.

First, any complex sentence has a *finite* number of basic sentences as components. It is thus possible simply to *list* all the possible interpretations of those basic sentences. If an expression has just one basic sentence $\mathcal{A}$, then on any interpretation whatsoever, that basic sentence must be T or F.

(D)

$\mathcal{A}$
T
F

If an expression has basic sentences $\mathcal{A}$ and $\mathcal{B}$, then the possible interpretations of its basic sentences are,

(E)

$\mathcal{A}$	$\mathcal{B}$
T	T
T	F
F	T
F	F

$\mathcal{B}$ can take its possible values, T and F when $\mathcal{A}$ is true, and $\mathcal{B}$ can take its possible values, T and F when $\mathcal{A}$ is false. And similarly, every time we add a basic sentence, we double the number of possible interpretations, so that n basic sentences always

have 2^n possible interpretations. Thus the possible interpretations for three and four basic sentences are,

(F)

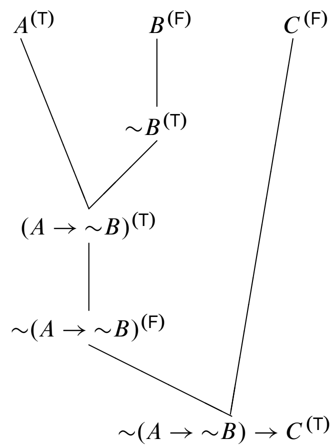
$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$
T	T	T
T	T	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

(G)

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$
T	T	T	T
T	T	T	F
T	T	F	T
T	T	F	F
T	F	T	T
T	F	T	F
T	F	F	T
T	F	F	F
F	T	T	T
F	T	T	F
F	T	F	T
F	T	F	F
F	F	T	T
F	F	T	F
F	F	F	T
F	F	F	F

Extra horizontal lines are added purely for visual convenience. There are $8 = 2^3$ combinations with three basic sentences and $16 = 2^4$ combinations with four. In general, to write down all the possible combinations for n basic sentences, begin by finding the total number $r = 2^n$ of combinations or rows. Then write down a column with half that many $(r/2)$ Ts and half that many $(r/2)$ Fs; then a column alternating half again as many $(r/4)$ Ts and Fs; and a column alternating half again as many $(r/8)$ Ts and Fs — continuing to the n^{th} column alternating groups of just one T and one F. Thus, for example, with four basic sentences, $r = 2^4 = 16$; so we begin with a column consisting of $r/2 = 8$ Ts and $r/2 = 8$ Fs; this is followed by a column alternating groups of 4 Ts and 4 Fs, a column alternating groups of 2 Ts and 2 Fs, and a column alternating groups of 1 T and 1 F. And similarly in other cases.

Given an expression involving, say, four basic sentences, we could imagine doing trees for each of the 16 possible interpretations. But, to exhibit truth values for each of the possible interpretations, we can reduce the amount of work a bit — or at least represent it in a relatively compact form. Suppose $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] = \mathsf{F}$, and $\mathsf{I}[C] = \mathsf{F}$, and consider a tree as in (B) from above, along with a "compressed" version of the same information.

(H) 

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	F	F	F	T	T	T	F	***T***	F

In the table on the right, we begin by simply listing the interpretation we will consider in the lefthand part: A is T, B is F and C is F. Then, under each basic sentence, we put its truth value, and for each formula, we list its truth value *under its main operator*. Notice that the calculation must proceed *precisely* as it does in the tree. It is because B is F, that we put T under the second $\sim$. It is because A is T and $\sim B$ is T that we put a T under the first $\to$. It is because $(A \to \sim B)$ is T that we put F under the first $\sim$. And it is because $\sim(A \to \sim B)$ is F and C is F that we put a T under the second $\to$. In effect, then, we work "down" through the tree, only in this compressed form. We might think of truth values from the tree as "squished" up into the one row. Because there is a T under its main operator, we conclude that the whole formula, $\sim(A \to \sim B) \to C$ is T when $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] = \mathsf{F}$, and $\mathsf{I}[C] = \mathsf{F}$. In this way, we might conveniently calculate and represent the truth value of $\sim(A \to \sim B) \to C$ for all eight of the possible interpretations of its basic sentences.

(I)

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	T	T	T	T	F	F	T	***T***	T
T	T	F	T	T	F	F	T	***F***	F
T	F	T	F	T	T	T	F	***T***	T
T	F	F	F	T	T	T	F	***T***	F
F	T	T	F	F	T	F	T	***T***	T
F	T	F	F	F	T	F	T	***T***	F
F	F	T	F	F	T	T	F	***T***	T
F	F	F	F	F	T	T	F	***T***	F

The emphasized column under the second $\to$ indicates the truth value of $\sim(A \to \sim B) \to C$ for each of the interpretations on the left — which is to say, for every possible interpretation of the three basic sentences. So the only way for $\sim(A \to \sim B) \to C$ to be F is for C to be F, and A and B to be T. Our above tree (H) represents just the fourth row of this table.

In practice, it is easiest to work these *truth tables* "vertically." For this, begin with the basic sentences in some standard order along with all their possible interpretations in the left-hand column. For $\mathcal{L}_s$ let the standard order be alphanumeric ($A, A_1, A_2 \ldots, B, B_1, B_2 \ldots, C \ldots$). And repeat truth values for basic sentences under their occurrences in the formula (this is not crucial, since truth values for basic sentences are already listed on the left; it will be up to you whether to repeat values for basic sentences). This is done in table (J) below.

(J)

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	T	T		T			T		T
T	T	F		T			T		F
T	F	T		T			F		T
T	F	F		T			F		F
F	T	T		F			T		T
F	T	F		F			T		F
F	F	T		F			F		T
F	F	F		F			F		F

(K)

A	B	C	$\sim$	$(A$	$\to$	$\sim$	$B)$	$\to$	C
T	T	T		T		F	T		T
T	T	F		T		F	T		F
T	F	T		T		T	F		T
T	F	F		T		T	F		F
F	T	T		F		F	T		T
F	T	F		F		F	T		F
F	F	T		F		T	F		T
F	F	F		F		T	F		F

Now, given the values for B as in (J), we are in a position to calculate the values for $\sim B$; so get the T($\sim$) table in you mind, put your eye on the column under B in the formula (or on the left if you have decided not to repeat the values for B under its occurrence in the formula). Then fill in the column under the second $\sim$, reversing the values from under B. This is accomplished in (K). Given the values for A and $\sim B$, we are now in a position to calculate values for $A \to \sim B$; so get the T($\to$) table in your head, and put your eye on the columns under A and $\sim B$. Then fill in the column

It is worth asking what happens if basic sentences are listed in some order other than alphanumeric.

A	B
T	T
T	F
F	T
F	F

B	A
T	T
T	F
F	T
F	F

All the combinations are still listed, but their locations in a table change.

Each of the above tables list all of the combinations for the basic sentences. But the first table has the interpretation I with I[A] = T and I[B] = F in the second row, where the second table has this combination in the third. Similarly, the tables exchange rows for the interpretation J with J[A] = F and J[B] = T. As it turns out, the only real consequence of switching rows is that it becomes difficult to compare tables as, for example, with the back of the book. And it may matter as part of the standard of correctness for exercises!

under the first →, going with F only when A is T and $\sim B$ is F. This is accomplished in (L).

(L)

A	B	C	$\sim$	$(A$	$\rightarrow$	$\sim$	$B)$	$\rightarrow$	C
T	T	T		T	F	F	T		T
T	T	F		T	F	F	T		F
T	F	T		T	T	T	F		T
T	F	F		T	T	T	F		F
F	T	T		F	T	F	T		T
F	T	F		F	T	F	T		F
F	F	T		F	T	T	F		T
F	F	F		F	T	T	F		F

(M)

A	B	C	$\sim$	$(A$	$\rightarrow$	$\sim$	$B)$	$\rightarrow$	C
T	T	T	T	T	F	F	T		T
T	T	F	T	T	F	F	T		F
T	F	T	F	T	T	T	F		T
T	F	F	F	T	T	T	F		F
F	T	T	F	F	T	F	T		T
F	T	F	F	F	T	F	T		F
F	F	T	F	F	T	T	F		T
F	F	F	F	F	T	T	F		F

Now we are ready to fill in the column under the first $\sim$. So get the T($\sim$) table in your head, and put your eye on the column under the first →. The column is completed in table (M). And the table is finished as in (I) by completing the column under the last →, based on the columns under the first $\sim$ and under the C. Notice again, that the order in which you work the columns exactly parallels the order from the tree.

As another example, consider these tables for $\sim(B \rightarrow A)$, the first with truth values repeated under basic sentences, the second without.

(N)

A	B	$\sim$	$(B$	$\rightarrow$	$A)$
T	T	***F***	T	T	T
T	F	***F***	F	T	T
F	T	***T***	T	F	F
F	F	***F***	F	T	F

(O)

A	B	$\sim$	$(B$	$\rightarrow$	$A)$
T	T	***F***		T	
T	F	***F***		T	
F	T	***T***		F	
F	F	***F***		T	

We complete the table as before. First, with our eye on the columns under B and A, we fill in the column under →. Then, with our eye on that column, we complete the one under $\sim$. For this, first, notice that $\sim$ is the *main* operator. You would *not* calculate $\sim B$ and then the arrow! Rather, your calculations move from the smaller parts to the larger; so the arrow comes first and then the tilde. Again, the order is the same as on a tree. Second, if you do not repeat values for basic formulas, be careful about $B \rightarrow A$; the leftmost column of table (O), under A, is the column for the *consequent* and the column immediately to its right, under B, is for the *antecedent*; in this case, then, the second row under arrow is T and the *third* is F. Though it is fine to omit columns under basic sentences, as they are already filled in on the left side, you should *not* skip other columns, as they are essential building blocks for the final result.

E4.2. For each of the following sentences of $\mathcal{L}_s$ construct a truth table to determine its truth value for each of the possible interpretations of its basic sentences.

*a. $\sim\sim A$

b. $\sim(A \rightarrow A)$

c. $(\sim A \rightarrow A)$

*d. $(\sim B \rightarrow A) \rightarrow B$

e. $\sim(B \rightarrow \sim A) \rightarrow B$

f. $(A \rightarrow \sim B) \rightarrow \sim(B \rightarrow \sim A)$

*g. $C \rightarrow (A \rightarrow B)$

h. $[A \rightarrow (C \rightarrow B)] \rightarrow [(A \rightarrow C) \rightarrow (A \rightarrow B)]$

*i. $(\sim A \rightarrow B) \rightarrow (\sim C \rightarrow D)$

j. $\sim(A \rightarrow \sim B) \rightarrow \sim(C \rightarrow \sim D)$

4.1.3 Validity

As we have seen, sentences are true and false relative to an interpretation. For any interpretation, a complex sentence has some definite value. But whether an argument is *sententially valid* depends on truth and falsity relative to *every* interpretation. Suppose a formal argument has premises $\mathcal{P}_1 \ldots \mathcal{P}_n$ and conclusion $\mathcal{Q}$. Then,

> $\mathcal{P}_1 \ldots \mathcal{P}_n$ *sententially entail* $\mathcal{Q}$ ($\mathcal{P}_1 \ldots \mathcal{P}_n \vDash_s \mathcal{Q}$) iff there is no sentential interpretation I such that $\mathsf{I}[\mathcal{P}_1] = \mathsf{T}$ and ... and $\mathsf{I}[\mathcal{P}_n] = \mathsf{T}$ but $\mathsf{I}[\mathcal{Q}] = \mathsf{F}$.

We can put this more generally as follows. Suppose Γ (Gamma) is a set of formulas, and say $\mathsf{I}[\Gamma] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ for each $\mathcal{P}$ in Γ. Then,

SV Γ *sententially entails* $\mathcal{Q}$ ($\Gamma \vDash_s \mathcal{Q}$) iff there is no sentential interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{Q}] = \mathsf{F}$.

Where the members of Γ are $\mathcal{P}_1 \ldots \mathcal{P}_n$, this says the same thing as before. Γ sententially entails $\mathcal{Q}$ when there is no sentential interpretation that makes each member of Γ true and $\mathcal{Q}$ false. If Γ sententially entails $\mathcal{Q}$ we say the *argument* whose premises are the members of Γ and conclusion is $\mathcal{Q}$ is *sententially valid*. Γ does not sententially entail $\mathcal{Q}$ ($\Gamma \nvDash_s \mathcal{Q}$) when there is some sentential interpretation on which all the members of Γ are true, but $\mathcal{Q}$ is false. We can think of the premises as *constraining* the interpretations that matter: for validity it is just the interpretations where the members of Γ are all true, on which the conclusion $\mathcal{Q}$ cannot be false. If Γ has no

members then there are no constraints on relevant interpretations, and the conclusion must be true on every interpretation in order for it to be valid. In this case, listing all the members of Γ individually, we simply write $\vDash_s \mathcal{Q}$, and if $\mathcal{Q}$ is valid, $\mathcal{Q}$ is *logically true* (a *tautology*). Notice the new double turnstile $\vDash$ for this semantic notion, in contrast to the single turnstile $\vdash$ for derivations from chapter 3.

Given that we are already in a position to exhibit truth values for arbitrary interpretations, it is a simple matter to determine whether an argument is sententially valid. Where the premises and conclusion of an argument include basic sentences $\mathcal{B}_1 \dots \mathcal{B}_n$, begin by calculating the truth values of the premises and conclusion for each of the possible interpretations for $\mathcal{B}_1 \dots \mathcal{B}_n$. Then *look* to see if any interpretation makes all the premises true but the conclusion false. If no interpretation makes the premises true and the conclusion not, then by SV, the argument is sententially valid. If some interpretation does make the premises true and the conclusion false, then it is not valid.

Thus, for example, suppose we want to know whether the following argument is sententially valid.

$$\text{(P)}\quad \begin{array}{l} (\sim A \rightarrow B) \rightarrow C \\ B \\ \hline C \end{array}$$

By SV, the question is whether there is an interpretation that makes the premises true and the conclusion not. So we begin by calculating the values of the premises and conclusion for each of the possible interpretations of the basic sentences in the premises and conclusion.

A	B	C	$(\sim$	A	$\rightarrow$	$B)$	$\rightarrow$	C	B	/	C
T	T	T	F	T	T	T	***T***	T	***T***		***T***
T	T	F	F	T	T	T	***F***	F	***T***		***F***
T	F	T	F	T	T	F	***T***	T	***F***		***T***
T	F	F	F	T	T	F	***F***	F	***F***		***F***
F	T	T	T	F	T	T	***T***	T	***T***		***T***
F	T	F	T	F	T	T	***F***	F	***T***		***F***
F	F	T	T	F	F	F	***T***	T	***F***		***T***
F	F	F	T	F	F	F	***T***	F	***F***		***F***

Now we simply look to see whether any interpretation makes all the premises true but the conclusion not. Interpretations represented by the top row, ones that make A, B, and C all T, do not make the premises true and the conclusion not, because both the premises and the conclusion come out true. In the second row, the conclusion is false, but the first premise is false as well; so not *all* the premises are true *and* the conclusion is false. In the third row, we do not have either all the premises true or the

conclusion false. In the fourth row, though the conclusion is false, the premises are not true. In the fifth row, the premises are true, but the conclusion is not false. In the sixth row, the first premise is not true, and in the seventh and eighth rows, the second premise is not true. So no interpretation makes the premises true and the conclusion false. So by SV, $(\sim A \to B) \to C, B \vDash_s C$. Notice that the only column that matters for a complex formula is the one under its main operator — the one that gives the value of *the sentence* for each of the interpretations; the other columns exist only to support the calculation of the value of the whole.

In contrast, $\sim[(B \to A) \to B] \nvDash_s \sim(A \to B)$. That is, an argument with premise, $\sim[(B \to A) \to B]$ and conclusion $\sim(A \to B)$ is not sententially valid.

(Q)

A	B	$\sim$	$[(B$	$\to$	$A)$	$\to$	$B]$	/	$\sim$	$(A$	$\to$	$B)$	
T	T	***F***	T	T	T	T	T		***F***	T	T	T	
T	F	***T***	F	T	T	F	F		***T***	T	F	F	
F	T	***F***	T	F	F	T	T		***F***	F	T	T	
F	F	***T***	F	T	F	F	F		***F***	F	T	F	$\Leftarrow$

In the first row, the premise is F. In the second, the conclusion is T. In the third, the premise is F. However, in the last, the premise is T, and the conclusion is F. So there are interpretations (any interpretation that makes A and B both F) that make the premise T and the conclusion not true. So by SV, $\sim[(B \to A) \to B] \nvDash_s \sim(A \to B)$, and the argument is not sententially valid. All it takes is *one* interpretation that makes all the premises T and the conclusion F to render an argument not sententially valid. Of course, there might be more than one, but one is enough!

As a final example, consider table (I) for $\sim(A \to \sim B) \to C$ on p. 101 above. From the table, there is an interpretation where the sentence is not true. Thus, by SV, $\nvDash_s \sim(A \to \sim B) \to C$. A sentence is valid only when it is true on every interpretation. Since there is an interpretation on which it is not true, the sentence is not valid (not a logical truth).

Since all it takes to demonstrate invalidity is *one* interpretation on which all the premises are true and the conclusion is false, we do not actually need an entire table to demonstrate invalidity. You may decide to produce a whole truth table in order to *find* an interpretation to demonstrate invalidity. But we can sometimes work "backward" from what we are trying to show to an interpretation that does the job. Thus, for example, to find the result from table (Q), we need an interpretation on which the premise is T and the conclusion is F. That is, we need a row like this,

(R)

A	B	$\sim$	$[(B$	$\to$	$A)$	$\to$	$B]$	/	$\sim$	$(A$	$\to$	$B)$
		T							***F***			

In order for the premise to be T, the conditional in the brackets must be F. And in order for the conclusion to be F, the conditional must be T. So we can fill in this

much.

(S)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
		T				F			***F***		T	

Since there are three ways for an arrow to be T, there is not much to be done with the conclusion. But since the conditional in the premise is F, we know that its antecedent is T and consequent is F. So we have,

(T)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
		T		T		F	F		***F***		T	

That is, if the conditional in the brackets is F, then $(B \rightarrow A)$ is T and B is F. But now we can fill in the information about B wherever it occurs. The result is as follows.

(U)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
	F	***T***	F	T		F	F		***F***		T	F

Since the first B in the premise is F, the first conditional in the premise is T irrespective of the assignment to A. But, with B false, the only way for the conditional in the argument's conclusion to be T is for A to be false as well. The result is our completed row.

(V)

A	B	$\sim$	$[(B$	$\rightarrow$	$A)$	$\rightarrow$	$B]$	/	$\sim$	$(A$	$\rightarrow$	$B)$
F	F	***T***	F	T	F	F	F		***F***	F	T	F

And we have recovered the row that demonstrates invalidity — without doing the entire table. In this case, the full table had only four rows, and we might just as well have done the whole thing. However, when there are many rows, this "shortcut" approach can be attractive. A disadvantage is that sometimes it is not obvious just *how* to proceed. In this example each stage led to the next. At stage (S), there were three ways to make the conclusion true. We were able to proceed insofar as the premise forced the next step. But it might have been that neither the premise nor the conclusion forced a definite next stage. In this sort of case, you might decide to do the whole table, just so that you can grapple with all the different combinations in an orderly way.

Notice what happens when we try this approach with an argument that is not invalid. Returning to argument (P) above, suppose we try to find a row where the premises are T and the conclusion is F. That is, we set out to find a row like this,

(W)

A	B	C	$(\sim$	A	$\rightarrow$	$B)$	$\rightarrow$	C	B	/	C
							T		***T***		***F***

Immediately, we are in a position to fill in values for B and C.

(X)

A	B	C	$(\sim$	A	$\rightarrow$	$B)$	$\rightarrow$	C	B	/	C
	T	F				T	***T***	F	***T***		***F***

Since the first premise is a true arrow with a false consequent, its antecedent $(\sim A \rightarrow B)$ must be F. But this requires that $\sim A$ be T and that B be F.

(Y)

A	B	C	$(\sim$	A	$\rightarrow$	$B)$	$\rightarrow$	C	B	/	C
	T	F	T		F	F/T	***T***	F	***T***		***F***

And there is no way to set B to F, as we have already seen that it has to be T in order to keep the second premise true — and no interpretation makes B *both* T and F. At this stage, we know, in our hearts, that there is no way to make both of the premises true and the conclusion false. In Part II we will turn this knowledge into an official mode of reasoning for validity. However, for now, let us consider a single row of a truth table (or a marked row of a full table) sufficient to demonstrate *invalidity*, but require a full table, exhibiting all the options, to show that an argument is sententially valid.

You may encounter odd situations where premises are never T, where conclusions are never F, or whatever. But if you stick to the definition, always asking whether there is any interpretation of the basic sentences that makes all the premises T and the conclusion F, all will be well.

E4.3. For each of the following, use truth tables to decide whether the entailment claims hold. Notice that a couple of the tables are already done from E4.2.

*a. $A \rightarrow \sim A \vDash_s \sim A$

b. $\sim A \rightarrow A \vDash_s \sim A$

*c. $A \rightarrow B, \sim A \vDash_s \sim B$

d. $A \rightarrow B, \sim B \vDash_s \sim A$

e. $\sim(A \rightarrow \sim B) \vDash_s B$

f. $\vDash_s C \rightarrow (A \rightarrow B)$

*g. $\vDash_s [A \rightarrow (C \rightarrow B)] \rightarrow [(A \rightarrow C) \rightarrow (A \rightarrow B)]$

h. $(A \rightarrow B) \rightarrow \sim(B \rightarrow A), \sim A, \sim B \vDash_s \sim(C \rightarrow C)$

i. $[A \rightarrow \sim(B \rightarrow \sim C)], [B \rightarrow (\sim C \rightarrow D)] \vDash_s A \rightarrow \sim(B \rightarrow \sim D)$

j. $\sim[(A \rightarrow \sim(B \rightarrow \sim C)) \rightarrow D], \sim D \rightarrow A \vDash_s C$

4.1.4 Abbreviations

We turn, finally to applications for our abbreviations. Consider, first, a truth table for $\mathscr{P} \vee \mathscr{Q}$, that is for $\sim\mathscr{P} \rightarrow \mathscr{Q}$.

T($\vee$)

$\mathscr{P}$	$\mathscr{Q}$	$\sim$	$\mathscr{P}$	$\rightarrow$	$\mathscr{Q}$
T	T	F	T	***T***	T
T	F	F	T	***T***	F
F	T	T	F	***T***	T
F	F	T	F	***F***	F

so that

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \vee \mathscr{Q}$
T	T	***T***
T	F	***T***
F	T	***T***
F	F	***F***

When $\mathscr{P}$ is T and $\mathscr{Q}$ is T, $\mathscr{P} \vee \mathscr{Q}$ is T; when $\mathscr{P}$ is T and $\mathscr{Q}$ is F, $\mathscr{P} \vee \mathscr{Q}$ is T; and so forth. Thus, when $\mathscr{P}$ is T and $\mathscr{Q}$ is T, we *know* that $\mathscr{P} \vee \mathscr{Q}$ is T, without going through all the steps to get there in the unabbreviated form. Just as when $\mathscr{P}$ is a formula and $\mathscr{Q}$ is a formula, we move directly to the conclusion that $\mathscr{P} \vee \mathscr{Q}$ is a formula without explicitly working all the intervening steps, so if we know the truth value of $\mathscr{P}$ and the truth value of $\mathscr{Q}$, we can move in a tree by the above table to the truth value of $\mathscr{P} \vee \mathscr{Q}$ without all the intervening steps. And similarly for the other abbreviating sentential operators. For $\wedge$,

T($\wedge$)

$\mathscr{P}$	$\mathscr{Q}$	$\sim$	$(\mathscr{P}$	$\rightarrow$	$\sim$	$\mathscr{Q})$
T	T	***T***	T	F	F	T
T	F	***F***	T	T	T	F
F	T	***F***	F	T	F	T
F	F	***F***	F	T	T	F

so that

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \wedge \mathscr{Q}$
T	T	***T***
T	F	***F***
F	T	***F***
F	F	***F***

T($\leftrightarrow$)

$\mathscr{P}$	$\mathscr{Q}$	$\sim$	$[(\mathscr{P}$	$\rightarrow$	$\mathscr{Q})$	$\rightarrow$	$\sim$	$(\mathscr{Q}$	$\rightarrow$	$\mathscr{P})]$
T	T	***T***	T	T	T	F	F	T	T	T
T	F	***F***	T	F	F	T	F	F	T	T
F	T	***F***	F	T	T	T	T	T	F	F
F	F	***T***	F	T	F	F	F	F	T	F

so that

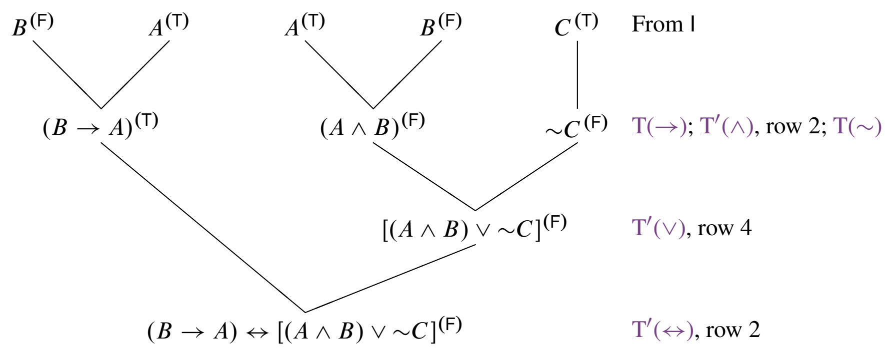
$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \leftrightarrow \mathscr{Q}$
T	T	***T***
T	F	***F***
F	T	***F***
F	F	***T***

As a help toward remembering these tables, notice that $\mathscr{P} \vee \mathscr{Q}$ is F only when $\mathscr{P}$ is F and $\mathscr{Q}$ is F; $\mathscr{P} \wedge \mathscr{Q}$ is T only when $\mathscr{P}$ is T and $\mathscr{Q}$ is T; and $\mathscr{P} \leftrightarrow \mathscr{Q}$ is T only when $\mathscr{P}$ and $\mathscr{Q}$ are the *same* and F when $\mathscr{P}$ and $\mathscr{Q}$ are different. We can think of these clauses as representing derived clauses T$'$($\vee$), T$'$($\wedge$), and T$'$($\leftrightarrow$) to the definition for truth.

And nothing prevents direct application of the derived tables in trees. Suppose, for example, I$[A]$ = T, I$[B]$ = F, and I$[C]$ = T. Then I$[(B \rightarrow A) \leftrightarrow [(A \wedge B) \vee \sim C]$ = F.

There are a couple of different ways tables for our operators can be understood: First, as we shall see in Part III, it is possible to take tables for operators other than $\sim$ and $\rightarrow$ as basic, say, just T($\sim$) and T($\vee$), or just T($\sim$) and T($\wedge$), and then abbreviate $\rightarrow$ in terms of them. Challenge: What expression involving just $\sim$ and $\vee$ has the same table as $\rightarrow$? what expression involving just $\sim$ and $\wedge$? Another option is to introduce all five as basic. Then the task is not *showing* that the table for $\vee$ is TTTF — that is given; rather we simply notice that $\mathcal{P} \vee \mathcal{Q}$, say, is redundant with $\sim\mathcal{P} \rightarrow \mathcal{Q}$. Again, our approach with $\sim$ and $\rightarrow$ basic has the advantage of preserving relative simplicity in the basic language (though other minimal approaches would do so as well).

(Z)



We might get the same result by working through the full tree for the unabbreviated form. But there is no need. When A is T and B is F, we *know* that $(A \wedge B)$ is F; when $(A \wedge B)$ is F and $\sim C$ is F, we *know* that $[(A \wedge B) \vee C]$ is F; and so forth. Thus we move through the tree directly by the derived tables.

Similarly, we can work directly with abbreviated forms in truth tables.

(AA)

A	B	C	$(B$	$\rightarrow$	$A)$	$\leftrightarrow$	$[(A$	$\wedge$	$B)$	$\vee$	$\sim$	$C]$
T	T	T	T	T	T	***T***	T	T	T	T	F	T
T	T	F	T	T	T	***T***	T	T	T	T	T	F
T	F	T	F	T	T	***F***	T	F	F	F	F	T
T	F	F	F	T	T	***T***	T	F	F	T	T	F
F	T	T	T	F	F	***T***	F	F	T	F	F	T
F	T	F	T	F	F	***F***	F	F	T	T	T	F
F	F	T	F	T	F	***F***	F	F	F	F	F	T
F	F	F	F	T	F	***T***	F	F	F	T	T	F

Tree (Z) represents just the third row of this table. As before, we construct the table "vertically," with tables for abbreviating operators in mind as appropriate.

Finally, given that we have tables for abbreviated forms, we can use them for evaluation of *arguments* with abbreviated forms. Thus, for example, $A \leftrightarrow B$, $A \vDash_s$

$A \wedge B$.

(AB)

A	B	$(A$	$\leftrightarrow$	$B)$	A	/	$(A$	$\wedge$	$B)$
T	T	T	***T***	T	***T***		T	***T***	T
T	F	T	***F***	F	***T***		T	***F***	F
F	T	F	***F***	T	***F***		F	***F***	T
F	F	F	***T***	F	***F***		F	***F***	F

There is no row where each of the premises is true and the conclusion is false. So the argument is sententially valid. And, from either of the following rows,

(AC)

A	B	C	D	$[(B$	$\rightarrow$	$A)$	$\wedge$	$(\sim$	C	$\vee$	$D)]$	$[(A$	$\leftrightarrow$	$\sim$	$D)$	$\wedge$	$(\sim$	D	$\rightarrow$	$B)]$	/	B
F	F	T	T	F	T	F	***T***	F	T	T	T	F	T	F	T	***T***	F	T	T	F		***F***
F	F	F	T	F	T	F	***T***	T	F	T	T	F	T	F	T	***T***	F	T	T	F		***F***

we may conclude that $[(B \rightarrow A) \wedge (\sim C \vee D)]$, $[(A \leftrightarrow \sim D) \wedge (\sim D \rightarrow B)] \not\models_s B$. In this case, the shortcut table is attractive relative to the full version with sixteen rows!

E4.4. For each of the following, use truth tables to decide whether the entailment claims hold.

a. $\models_s A \vee \sim A$

b. $A \leftrightarrow [\sim A \leftrightarrow (A \wedge \sim A)]$, $A \rightarrow \sim(A \leftrightarrow A) \models_s \sim A \rightarrow A$

*c. $B \vee \sim C \models_s B \rightarrow C$

*d. $A \vee B$, $\sim C \rightarrow \sim A$, $\sim(B \wedge \sim C) \models_s C$

e. $A \rightarrow (B \vee C)$, $C \leftrightarrow B$, $\sim C \models_s \sim A$

f. $\sim(A \wedge \sim B) \models_s \sim A \vee B$

g. $A \wedge (B \rightarrow C) \models_s (A \wedge B) \vee (A \wedge C)$

*h. $\models_s \sim(A \leftrightarrow B) \leftrightarrow (A \wedge \sim B)$

i. $A \vee (B \wedge \sim C)$, $\sim(\sim B \vee C) \rightarrow \sim A \models_s \sim A \leftrightarrow \sim(C \vee \sim B)$

j. $A \vee B$, $\sim D \rightarrow (C \vee A) \models_s B \leftrightarrow \sim C$

E4.5. For each of the following, use truth tables to decide whether the entailment claims hold. Hint: the trick here is to identify the *basic* sentences. After that, everything proceeds in the usual way with truth values assigned to the basic sentences.

Semantics Quick Reference (Sentential)

For any formal language $\mathcal{L}$, a *sentential interpretation* assigns a truth value *true* or *false*, T or F, to each of its basic sentences. Then for any interpretation I,

ST ($\sim$) For any sentence $\mathcal{P}$, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{F}$; otherwise $\mathsf{I}[\sim\mathcal{P}] = \mathsf{F}$.

($\rightarrow$) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, $\mathsf{I}[(\mathcal{P} \rightarrow \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{F}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$ (or both); otherwise $\mathsf{I}[(\mathcal{P} \rightarrow \mathcal{Q})] = \mathsf{F}$.

And for abbreviated expressions,

ST′ ($\wedge$) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, $\mathsf{I}[(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$; otherwise $\mathsf{I}[(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{F}$.

($\vee$) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, $\mathsf{I}[(\mathcal{P} \vee \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$ (or both); otherwise $\mathsf{I}[(\mathcal{P} \vee \mathcal{Q})] = \mathsf{F}$.

($\leftrightarrow$) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, $\mathsf{I}[(\mathcal{P} \leftrightarrow \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{Q}]$; otherwise $\mathsf{I}[(\mathcal{P} \leftrightarrow \mathcal{Q})] = \mathsf{F}$.

If Γ (Gamma) is a set of formulas, $\mathsf{I}[\Gamma] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ for each $\mathcal{P}$ in Γ. Then, where the members of Γ are the formal premises of an argument, and sentence $\mathcal{P}$ is its conclusion,

SV Γ *sententially entails* $\mathcal{P}$ iff there is no sentential interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{P}] = \mathsf{F}$.

We treat a single row of a truth table (or a marked row of a full table) as sufficient to demonstrate *invalidity*, but require a full table, exhibiting all the options, to show that an argument is sententially valid.

*a. $\exists x Ax \rightarrow \exists x Bx, \sim\exists x Ax \vDash_s \exists x Bx$

b. $\forall x Ax \rightarrow \sim\exists x(Ax \wedge \forall y By), \exists x(Ax \wedge \forall y By) \vDash_s \sim\forall x Ax$

E4.6. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Sentential interpretations and truth for complex sentences.

b. Sentential validity.

4.2 Quantificational

Semantics for the quantificational case work along the same lines as the sentential one. Sentences are true or false relative to an interpretation; arguments are semantically valid when there is no interpretation on which the premises are true and the conclusion is not. But, corresponding to differences between sentential and quantificational languages, the notion of an interpretation differs. And we introduce a preliminary notion of a *term* assignment, along with a preliminary notion of *satisfaction* distinct from truth, before we get to truth and validity. Certain issues are put off for chapter 7 at the start of Part II. However, we should be able to do enough to see *how* the definitions work. This time, we will say a bit more about connections to English, though it remains important to see the definitions for what they are, and we leave official discussion of translation to the next chapter.

4.2.1 Interpretations

Given a quantificational language $\mathscr{L}$, formulas are true relative to a *quantificational interpretation*. As in the sentential case, languages do not *come* associated with any interpretation. Rather, a language consists of symbols which may be interpreted in different ways. In the sentential case, interpretations assigned T or F to basic sentences — and the assignments were made in arbitrary ways. Now assignments are more complex, but remain arbitrary. In general,

QI A *quantificational interpretation* I of language $\mathscr{L}$, consists of a nonempty set U, the *universe* of the interpretation, along with,

(s) An assignment of a truth value $\mathsf{I}[\mathcal{S}]$ to each sentence letter $\mathcal{S}$ of $\mathscr{L}$.

(c) An assignment of a member $\mathsf{I}[c]$ of U to each constant symbol c of $\mathscr{L}$.

(r) An assignment of an n-place relation $\mathsf{I}[\mathcal{R}^n]$ on U to each n-place relation symbol $\mathcal{R}^n$ of $\mathscr{L}$, where $\mathsf{I}[=]$ is always assigned $\{\langle \mathsf{o}, \mathsf{o}\rangle \mid \mathsf{o} \in \mathsf{U}\}$.

(f) An assignment of a total n-place function $\mathsf{I}[h^n]$ from U^n to U, to each n-place function symbol h^n of $\mathscr{L}$.

The notions of a *function* and a *relation* come from set theory, for which you might want to check out the set theory summary on p. 114. Conceived literally and mathe-

Basic Notions of Set Theory

I. A *set* is a thing that may have other things as elements or members. If m is a member of set s we write $\mathsf{m} \in \mathsf{s}$. One set is identical to another iff their members are the same — so order is irrelevant. The members of a set may be specified by list: {Sally, Bob, Jim}, or by membership condition: $\{\mathsf{o} \mid \mathsf{o}$ is a student at CSUSB$\}$; read, 'the set of all objects o such that o is a student at CSUSB'. Since sets are things, nothing prevents a set with other sets as members.

II. Like a set, an *n-tuple* is a thing with other things as elements or members. For any integer n, an n-tuple has n elements, where order matters. 2-tuples are frequently referred to as "pairs." An n-tuple may be specified by list: ⟨Sally, Bob, Jim⟩, or by membership condition, 'the first 5 people (taken in order) in line at the Bursar's window'. Nothing prevents sets of n-tuples, as $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}$ loves $\mathsf{n}\}$; read, 'the set of all m/n pairs such that the first member loves the second'. 1-tuples are frequently equated with their members. So, depending on context, {Sally, Bob, Jim} may be {⟨Sally⟩, ⟨Bob⟩, ⟨Jim⟩}.

III. Set r is a *subset* of set s iff any member of r is also a member of s. If r is a subset of s we write $\mathsf{r} \subseteq \mathsf{s}$. r is a *proper subset* of s ($\mathsf{r} \subset \mathsf{s}$) iff $\mathsf{r} \subseteq \mathsf{s}$ but $\mathsf{r} \neq \mathsf{s}$. Thus, for example, the subsets of $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$ are $\{\}$, $\{\mathsf{m}\}$, $\{\mathsf{n}\}$, $\{\mathsf{o}\}$, $\{\mathsf{m}, \mathsf{n}\}$, $\{\mathsf{m}, \mathsf{o}\}$, $\{\mathsf{n}, \mathsf{o}\}$, and $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$. All but $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$ are proper subsets of $\{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$. Notice that the *empty set* is a subset of any set s, for it is sure to be the case that any member of it is also a member of s.

IV. The *union* of sets r and s is the set of all objects that are members of r *or* s. Thus, if $\mathsf{r} = \{\mathsf{m}, \mathsf{n}\}$ and $\mathsf{s} = \{\mathsf{n}, \mathsf{o}\}$, then the union of r and s, $(\mathsf{r} \cup \mathsf{s}) = \{\mathsf{m}, \mathsf{n}, \mathsf{o}\}$. Given a larger collection of sets, s_1, $\mathsf{s}_2 \ldots$ the union of them all, $\bigcup \mathsf{s}_1$, $\mathsf{s}_2 \ldots$ is the set of all objects that are members of s_1, or s_2, or $\ldots$. Similarly, the *intersection* of sets r and s is the set of all objects that are members of r *and* s. Thus the intersection of r and s, $(\mathsf{r} \cap \mathsf{s}) = \{\mathsf{n}\}$, and $\bigcap \mathsf{s}_1, \mathsf{s}_2 \ldots$ is the set of all objects that are members of s_1, and s_2, and $\ldots$.

V. Let s^{n} be the set of all n-tuples formed from members of s. Then an n-place *relation on set* s is any subset of s^{n}. Thus, for example, $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}$ is married to $\mathsf{n}\}$ is a subset of the pairs of people, and so is a 2-place relation on the set of people. An *n-place function from* r^{n} *to* s is a set of pairs whose first member is an element of r^{n} and whose second member is an element of s — where no member of r^{n} is paired with more than one member of s. Thus $\langle\langle 1, 1\rangle, 2\rangle$ and $\langle\langle 1, 2\rangle, 3\rangle$ might be members of an addition function. $\langle\langle 1, 1\rangle, 2\rangle$ and $\langle\langle 1, 1\rangle, 3\rangle$ could not be members of the *same* function. A *total* function from r^{n} to s is one that pairs *each* member of r^{n} with some member of s. We think of the first element of these pairs as an *input*, and the second as the function's *output* for that input. Thus if $\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \in \mathsf{f}$ we say $\mathsf{f}(\mathsf{m}, \mathsf{n}) = \mathsf{o}$.

matically, these assignments are themselves *functions* from symbols in the language $\mathscr{L}$ to objects. Each sentence letter is associated with a truth value, T or F — this is no different than before. Each constant symbol is associated with some element of U. Each n-place relation symbol is associated with a subset of U^{n} — with a set whose members are of the sort $\langle \mathsf{a}_1 \ldots \mathsf{a}_{\mathsf{n}} \rangle$ where $\mathsf{a}_1 \ldots \mathsf{a}_{\mathsf{n}}$ are elements of U. And each n-place function symbol is associated with a set whose members are of the sort $\langle\langle \mathsf{a}_1 \ldots \mathsf{a}_{\mathsf{n}} \rangle, \mathsf{b}\rangle$ where $\mathsf{a}_1 \ldots \mathsf{a}_{\mathsf{n}}$ and b are elements of U. And where $\mathsf{U} = \{\mathsf{a}, \mathsf{b}, \mathsf{c} \ldots\}$, $\mathsf{I}[=]$ is $\{\langle \mathsf{a}, \mathsf{a}\rangle, \langle \mathsf{b}, \mathsf{b}\rangle, \langle \mathsf{c}, \mathsf{c}\rangle, \ldots\}$. Notice that U may be any non-empty set, and so need not be countable. *Any* such assignments count as a quantificational interpretation.

Intuitively, the universe contains whatever objects are under consideration in a given context. Thus one may ask whether "everyone" understands the notion of an interpretation, and have in mind some limited collection of individuals — not literally everyone. Constant symbols work like proper names: Constant symbol a names the object $\mathsf{I}[a]$ with which it is associated. So, for example, in $\mathscr{L}_q$ we might set $\mathsf{I}[b]$ to Bill, and $\mathsf{I}[h]$ to Hillary. Relation symbols are interpreted like predicates: Relation symbol $\mathscr{R}^n$ *applies* to the n-tuples with which it is associated. Thus, in $\mathscr{L}_q$ where U is the set of all people, we might set $\mathsf{I}[H^1]$ to $\{\mathsf{o} \mid \mathsf{o}$ is happy$\}$,[2] and $\mathsf{I}[L^2]$ to $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}$ loves $\mathsf{n}\}$. Then if Bill is happy, H applies to Bill, and if Bill loves Hillary, L applies to ⟨Bill, Hillary⟩, though if she is mad enough, L might not apply to ⟨Hillary, Bill⟩. Function symbols are used to pick out one object by means of other(s). Thus, when we say that *Bill's father* is happy, we pick out an object (the father) by means of another (Bill). Similarly, function symbols are like "oblique" names which pick out objects in response to inputs. Such behavior is commonplace in mathematics when we say, for example that 3 + 3 is even — and we are talking about 6. Thus we might assign $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{n}$ is the father of $\mathsf{m}\}$ to one-place function symbol f and $\{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}$ plus $\mathsf{n} = \mathsf{o}\}$ to two-place function symbol p.

For some examples of interpretations, let us return to the language $\mathscr{L}_{\mathrm{NT}}^{<}$ from section 2.2.5 on p. 63. Recall that $\mathscr{L}_{\mathrm{NT}}^{<}$ includes just constant symbol $\emptyset$; two-place relation symbols $<, =$; one-place function symbol S; and two-place function symbols $\times$ and $+$. Given these symbols, terms and formulas are generated in the usual way. Where $\mathbb{N}$ is the set $\{0, 1, 2 \ldots\}$ of *natural numbers*[3] and the *successor* of any integer is the integer after it, the *standard* interpretation $\mathsf{N1}$ for $\mathscr{L}_{\mathrm{NT}}^{<}$ has universe $\mathbb{N}$ with,

---

[2]Or $\{\langle \mathsf{o}\rangle \mid \mathsf{o}$ is happy $\}$. As mentioned in the set theory guide, one-tuples are collapsed into their members.

[3]There is a problem of terminology: Strangely, many texts for elementary and high school mathematics exclude zero from the natural numbers, where most higher-level texts do not. We take the latter course.

$$\begin{array}{ll} \mathsf{N1} & \mathsf{N1}[\emptyset] = 0 \\ & \mathsf{N1}[<] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ is less than } \mathsf{n}\} \\ & \mathsf{N1}[S] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}, \text{ and } \mathsf{n} \text{ is the successor of } \mathsf{m}\} \\ & \mathsf{N1}[+] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ plus } \mathsf{n} \text{ equals } \mathsf{o}\} \\ & \mathsf{N1}[\times] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ times } \mathsf{n} \text{ equals } \mathsf{o}\} \end{array}$$

where it is automatic from QI that $\mathsf{N1}[=]$ is $\{\langle 1, 1\rangle, \langle 2, 2\rangle, \langle 3, 3\rangle \ldots\}$. The *standard interpretation* N of the minimal language $\mathcal{L}_{\mathrm{NT}}$ which omits the $<$ symbol is like $\mathsf{N1}$ but without the interpretation of $<$. These definitions work just as we expect. Thus, for example,

$$\text{(AD)} \quad \begin{array}{l} \mathsf{N1}[S] = \{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle \ldots\} \\ \mathsf{N1}[<] = \{\langle 0, 1\rangle, \langle 0, 2\rangle, \langle 0, 3\rangle, \ldots \langle 1, 2\rangle, \langle 1, 3\rangle \ldots\} \\ \mathsf{N1}[+] = \{\langle\langle 0, 0\rangle, 0\rangle, \langle\langle 0, 1\rangle, 1\rangle, \langle\langle 0, 2\rangle, 2\rangle, \ldots \langle\langle 1, 0\rangle, 0\rangle, \langle\langle 1, 1\rangle, 2\rangle, \ldots\} \end{array}$$

The standard interpretation represents the way you have understood these symbols since grade school.

But there is nothing sacred about this interpretation. Thus, for example, we might introduce an I with U = {Bill, Hill} and,

$$\begin{array}{ll} \mathsf{I} & \mathsf{I}[\emptyset] = \text{Bill} \\ & \mathsf{I}[<] = \{\langle \text{Hill, Hill}\rangle, \langle \text{Hill, Bill}\rangle\} \\ & \mathsf{I}[S] = \{\langle \text{Bill, Bill}\rangle, \langle \text{Hill, Hill}\rangle\} \\ & \mathsf{I}[+] = \{\langle\langle \text{Bill, Bill}\rangle, \text{Bill}\rangle, \langle\langle \text{Bill, Hill}\rangle, \text{Bill}\rangle, \langle\langle \text{Hill, Bill}\rangle, \text{Hill}\rangle, \langle\langle \text{Hill, Hill}\rangle, \text{Hill}\rangle\} \\ & \mathsf{I}[\times] = \{\langle\langle \text{Bill, Bill}\rangle, \text{Hill}\rangle, \langle\langle \text{Bill, Hill}\rangle, \text{Bill}\rangle, \langle\langle \text{Hill, Bill}\rangle, \text{Bill}\rangle, \langle\langle \text{Hill, Hill}\rangle, \text{Bill}\rangle\} \end{array}$$

This assigns a member of the universe to the constant symbol; a set of pairs to the two-place relation symbol (where the interpretation of $=$ is automatic); a total 1-place function to S, and total 2-place functions to $\times$ and $+$. So it counts as an interpretation of $\mathcal{L}^{<}_{\mathrm{NT}}$.

It is frequently convenient to link assignments with bits of (relatively) ordinary language. This is a key to translation, as explored in the next chapter! But there is no requirement that we link up with ordinary language. All that is required is that we assign a member of U to the constant symbol, a subset of U^2 to the 2-place relation symbol, and a total function from U^{n} to U to each n-place function symbol. That is all that is required — and nothing beyond that is required in order to say what the function and predicate symbols "mean." So I counts as a legitimate (though non-standard) interpretation of $\mathcal{L}^{<}_{\mathrm{NT}}$. With a language like $\mathcal{L}_q$ it is not always possible to specify assignments for *all* the symbols in the language. Even so, we can specify a *partial* interpretation — an interpretation for the symbols that matter in a given context.

E4.7. Suppose Bill and Hill have another child and (for reasons known only to them) name him Dill. Where U = {Bill, Hill, Dill}, give another interpretation J for $\mathcal{L}^{<}_{NT}$. Arrange your interpretation so that: (i) J[∅] ≠ Bill; (ii) there are exactly five pairs in J[<]); and (iii) for any m, ⟨⟨m, Bill⟩, Dill⟩ and ⟨⟨Bill, m⟩, Dill⟩ are in J[+]. Include J[=] in your account. Hint: a two-place total function on a three-member universe should have $3^2 = 9$ members.

4.2.2 Term Assignments

For some language $\mathcal{L}$, say U = {o | o is a person}, one-place predicate H is assigned the set of happy people, and constant b is assigned Bill. Perhaps H applies to Bill. In this case, Hb comes out true. Intuitively, however, we cannot say that Hx is either true or false on this interpretation, precisely because there is no particular individual that x picks out — we do not know *who* is supposed to be happy. However we will be able to say that Hx is *satisfied* or not when the interpretation is supplemented with a *variable (designation) assignment* d associating each variable with some individual in U.

Given a language $\mathcal{L}$ and interpretation I, a *variable assignment* d is a total function from the variables of $\mathcal{L}$ to objects in the universe U. Conceived pictorially, where U = $\{o_1, o_2 \ldots\}$, d and h are variable assignments.

$$\mathsf{d}\quad \begin{array}{ccccccccc} i & j & k & l & m & n & o & p & \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \ldots \\ o_1 & o_2 & o_3 & o_4 & o_5 & o_6 & o_7 & o_8 & \end{array}$$

$$\mathsf{h}\quad \begin{array}{cccccccccc} i & j & & k & l & m & n & o & p & \\ \downarrow & \downarrow & \swarrow & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \ldots \\ o_1 & o_2 & & o_3 & o_4 & o_5 & o_6 & o_7 & o_8 & \end{array}$$

If d assigns o to x we write d$[x]$ = o. So d$[k]$ = o_3 and h$[k]$ = o_2. Observe that the total function from variables to things assigns some element of U to every variable of $\mathcal{L}$. But this leaves room for one thing assigned to different variables, and things assigned to no variable at all. For any assignment d, d$(x|$o$)$ is the assignment that is just like d except that x is assigned to o. Thus, d$(k|o_2)$ = h. Similarly,

$$\mathsf{k}\quad \begin{array}{ccccccccccc} i & j & & k & l & & m & n & o & p & \\ \downarrow & \downarrow & \swarrow & & & \searrow & \downarrow & \downarrow & \downarrow & \downarrow & \ldots \\ o_1 & o_2 & & o_3 & o_4 & & o_5 & o_6 & o_7 & o_8 & \end{array}$$

$\mathsf{d}(k|\mathsf{o}_2, l|\mathsf{o}_5) = \mathsf{h}(l|\mathsf{o}_5) = \mathsf{k}$. Of course, if some d already has x assigned to o, then $\mathsf{d}(x|\mathsf{o})$ is just d. Thus, for example, $\mathsf{k}(i|\mathsf{o}_1)$ is just k itself. We will be willing to say that Hx is *satisfied* or not satisfied relative to an interpretation supplemented by a variable assignment. But before we get to satisfaction, we need the general notion of a *term* assignment.

In general, a term contributes to a formula by picking out some member of the universe U — terms act something like names. We have seen that an interpretation I assigns a member $\mathsf{I}[c]$ of U to each constant symbol c. And a variable assignment d assigns a member $\mathsf{d}[x]$ to each variable x. But these are assignments just to "basic" terms. An interpretation assigns to function symbols, not members of U, but certain complex sets. Still an interpretation I supplemented by a variable assignment d, is sufficient to associate a member $\mathsf{I}_\mathsf{d}[t]$ of U with any term t of $\mathscr{L}$. Where $\langle\langle \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}\rangle, \mathsf{b}\rangle \in \mathsf{I}[h^n]$, let $\mathsf{I}[h^n]\langle \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}\rangle = \mathsf{b}$; that is, $\mathsf{I}[h^n]\langle \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}\rangle$ is the thing the function $\mathsf{I}[h^n]$ associates with input $\langle \mathsf{a}_1 \ldots \mathsf{a}_\mathsf{n}\rangle$. Thus, for example, $\mathsf{N1}[+]\langle 1, 1\rangle = 2$ and $\mathsf{I}[+]\langle \text{Bill, Hill}\rangle = \text{Bill}$. Then for any interpretation I, variable assignment d, and term t,

TA (c) If c is a constant, then $\mathsf{I}_\mathsf{d}[c] = \mathsf{I}[c]$.

(v) If x is a variable, then $\mathsf{I}_\mathsf{d}[x] = \mathsf{d}[x]$.

(f) If h^n is a function symbol and $t_1 \ldots t_n$ are terms, then $\mathsf{I}_\mathsf{d}[h^n t_1 \ldots t_n] = \mathsf{I}[h^n]\langle \mathsf{I}_\mathsf{d}[t_1] \ldots \mathsf{I}_\mathsf{d}[t_n]\rangle$.

The first two clauses take over assignments to constants and variables from I and d. The last clause is parallel to the one by which terms are formed. The assignment to a complex term depends on assignments to the terms that are its parts, with the interpretation of the relevant function symbol. Again, the definition is recursive, and we can see how it works on a tree — in this case, one with the very same shape as the one by which we see that an expression is in fact a term. Say the interpretation of $\mathscr{L}^{<}_{\mathrm{NT}}$ is I as above, and $\mathsf{d}[x] = \text{Hill}$; then $\mathsf{I}_\mathsf{d}[S(Sx + \emptyset)] = \text{Hill}$.

(AE)

$x^{[\text{Hill}]}$ $\quad$ $\emptyset^{[\text{Bill}]}$ $\quad$ By TA(v) and TA(c)

$Sx^{[\text{Hill}]}$ $\quad$ With the input, since $\langle \text{Hill, Hill}\rangle \in \mathsf{I}[S]$, by TA(f)

$(Sx + \emptyset)^{[\text{Hill}]}$ $\quad$ With the inputs, since $\langle\langle \text{Hill, Bill}\rangle, \text{Hill}\rangle \in \mathsf{I}[+]$, by TA(f)

$S(Sx + \emptyset)^{[\text{Hill}]}$ $\quad$ With the input, since $\langle \text{Hill, Hill}\rangle \in \mathsf{I}[S]$, by TA(f)

As usual, basic elements occur in the top row. Other elements are fixed by ones that come before. The hard part about definition TA is just reading clause (f). It is perhaps easier to apply in practice than to read. For a complex term, assignments to terms that are the parts, together with the assignment to the function symbol determine the assignment to the whole. And this is just what clause (f) says. For practice, convince yourself that $\mathsf{I}_{\mathsf{d}(x|\text{Bill})}[S(Sx + \emptyset)] = \text{Bill}$, and where $\mathsf{N1}$ is as above and $\mathsf{d}[x] = 1$, $\mathsf{N1}_{\mathsf{d}}[S(Sx + \emptyset)] = 3$.

E4.8. For $\mathscr{L}_{NT}^{<}$ and interpretation $\mathsf{N1}$ as above on p. 115, let d include,

$$\begin{array}{ccccc} & w & x & y & z \\ \mathsf{d} & \downarrow & \downarrow & \downarrow & \downarrow \\ & 1 & 2 & 3 & 4 \end{array}$$

and use trees to determine each of the following.

*a. $\mathsf{N1}_{\mathsf{d}}[+xS\emptyset]$

b. $\mathsf{N1}_{\mathsf{d}}[x + (SS\emptyset \times x)]$

c. $\mathsf{N1}_{\mathsf{d}}[w \times S(\emptyset + (y \times SSSz))]$

*d. $\mathsf{N1}_{\mathsf{d}(x|4)}[x + (SS\emptyset \times x)]$

e. $\mathsf{N1}_{\mathsf{d}(x|1,w|2)}[S(x \times (S\emptyset + Sw))]$

E4.9. For $\mathscr{L}_{NT}^{<}$ and interpretation I as above on p. 116, let d include,

$$\begin{array}{ccccc} & w & x & y & z \\ \mathsf{d} & \downarrow & \downarrow & \downarrow & \downarrow \\ & \text{Bill} & \text{Hill} & \text{Hill} & \text{Hill} \end{array}$$

and use trees to determine each of the following.

*a. $\mathsf{I}_{\mathsf{d}}[+xS\emptyset]$

b. $\mathsf{I}_{\mathsf{d}}[x + (SS\emptyset \times x)]$

c. $\mathsf{I}_{\mathsf{d}}[w \times S(\emptyset + (y \times SSSz))]$

*d. $\mathsf{I}_{\mathsf{d}(x|\text{Bill})}[x + (SS\emptyset \times x)]$

e. $\mathsf{I}_{\mathsf{d}(x|\text{Bill},w|\text{Hill})}[S(x \times (S\emptyset + Sw))]$

E4.10. Consider your interpretation J for $\mathcal{L}_{NT}^{<}$ from E4.7. Supposing that $\mathsf{d}[w]$ = Bill, $\mathsf{d}[y]$ = Hill, and $\mathsf{d}[z]$ = Dill, determine $\mathsf{J_d}[w \times S(\emptyset + (y \times SSSz))]$. Explain how your interpretation has this result.

E4.11. For $\mathcal{L}_q$ and an interpretation K with universe U = {Amy, Bob, Chris} with,

$$
\begin{aligned}
\mathsf{K} \qquad & \mathsf{K}[a] = \text{Amy} \\
& \mathsf{K}[c] = \text{Chris} \\
& \mathsf{K}[f^1] = \{\langle \text{Amy, Bob}\rangle, \langle \text{Bob, Chris}\rangle, \langle \text{Chris, Amy}\rangle\} \\
& \mathsf{K}[g^2] = \{\langle\langle \text{Amy, Amy}\rangle, \text{Amy}\rangle, \langle\langle \text{Amy, Bob}\rangle, \text{Chris}\rangle, \langle\langle \text{Amy, Chris}\rangle, \text{Bob}\rangle, \langle\langle \text{Bob, Bob}\rangle, \text{Bob}\rangle, \langle\langle \text{Bob, Chris}\rangle, \text{Amy}\rangle, \langle\langle \text{Bob, Amy}\rangle, \text{Chris}\rangle, \langle\langle \text{Chris, Chris}\rangle, \text{Chris}\rangle, \langle\langle \text{Chris, Amy}\rangle, \text{Bob}\rangle, \langle\langle \text{Chris, Bob}\rangle, \text{Amy}\rangle\}
\end{aligned}
$$

where $\mathsf{d}(x)$ = Bob, $\mathsf{d}(y)$ = Amy and $\mathsf{d}(z)$ = Bob, use trees to determine each of the following,

a. $\mathsf{K_d}[f^1c]$

*b. $\mathsf{K_d}[g^2yf^1c]$

c. $\mathsf{K_d}[g^2g^2axf^1c]$

d. $\mathsf{K_{d(x|\text{Chris})}}[g^2g^2axf^1c]$

e. $\mathsf{K_{d(x|\text{Amy})}}[g^2g^2g^2xyzg^2f^1af^1c]$

4.2.3 Satisfaction

A term's assignment depends on an interpretation supplemented by an assignment for variables, that is, on some $\mathsf{I_d}$. Similarly, a formula's *satisfaction* depends on both the interpretation and variable assignment. As we shall see, however, *truth* is fixed by the interpretation I alone — just as in the sentential case. If a formula $\mathcal{P}$ is satisfied on I supplemented with d, we write $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$; if $\mathcal{P}$ is not satisfied on I with d, $\mathsf{I_d}[\mathcal{P}] = \mathsf{N}$. For any interpretation I with variable assignment d,

SF (s) If $\mathcal{S}$ is a sentence letter, then $\mathsf{I_d}[\mathcal{S}] = \mathsf{S}$ iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; otherwise $\mathsf{I_d}[\mathcal{S}] = \mathsf{N}$.

(r) If $\mathcal{R}^n$ is an n-place relation symbol and $t_1 \dots t_n$ are terms, $\mathsf{I_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$ iff $\langle \mathsf{I_d}[t_1] \dots \mathsf{I_d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; otherwise $\mathsf{I_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{N}$.

($\sim$) If $\mathcal{P}$ is a formula, then $\mathsf{I_d}[\sim\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{P}] = \mathsf{N}$; otherwise $\mathsf{I_d}[\sim\mathcal{P}] = \mathsf{N}$.

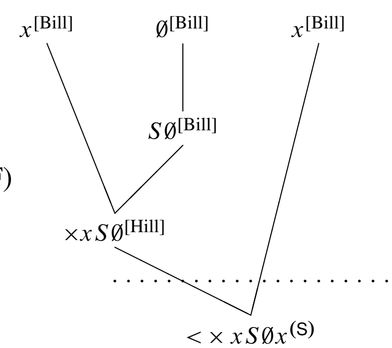
($\rightarrow$) If $\mathcal{P}$ and $\mathcal{Q}$ are formulas, then $\mathsf{I_d}[(\mathcal{P} \rightarrow \mathcal{Q})] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{P}] = \mathsf{N}$ or $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$ (or both); otherwise $\mathsf{I_d}[(\mathcal{P} \rightarrow \mathcal{Q})] = \mathsf{N}$.

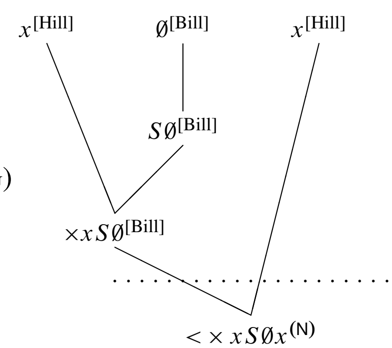
($\forall$) If $\mathcal{P}$ is a formula and x is a variable, then $\mathsf{I_d}[\forall x \mathcal{P}] = \mathsf{S}$ iff for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I_{d(x|o)}}[\mathcal{P}] = \mathsf{S}$; otherwise $\mathsf{I_d}[\forall x \mathcal{P}] = \mathsf{N}$.

SF(s), SF($\sim$) and SF($\rightarrow$) are closely related to ST from before, though satisfaction applies now to any *formulas* and not only to sentences. Other clauses are new.

SF(s) and SF(r) determine satisfaction for atomic formulas. Satisfaction for other formulas depends on satisfaction of their immediate subformulas. First, the satisfaction of a sentence letter works just like truth before: If a sentence letter is true on an interpretation, then it is satisfied. Thus satisfaction for sentence letters depends only on the interpretation, and not at all on the variable assignment.

In contrast, to see if $\mathcal{R}^n t_1 \dots t_n$ is satisfied, we find out which things are assigned to the terms. It is natural to think about this on a tree like the one by which we show that the expression is a formula. Thus given interpretation I for $\mathcal{L}_{NT}^{<}$ from p. 116, consider $(x \times S\emptyset) < x$; and compare cases with $\mathsf{d}[x] =$ Bill, and $\mathsf{h}[x] =$ Hill. It will be convenient to think about the expression in its unabbreviated form, $< \times x S \emptyset x$.

(AF) 

(AG) 

Above the dotted line, we calculate term assignments in the usual way. Assignment d is worked out on the left, and h on the right. But $< \times x S\emptyset x$ is a formula of the sort $<t_1 t_2$. From diagram (AF), $\mathsf{I_d}[\times x S\emptyset] =$ Hill, and $\mathsf{I_d}[x] =$ Bill. So the assignments to t_1 and t_2 are Hill and Bill. Since $\langle$Hill, Bill$\rangle \in \mathsf{I}[<]$, by SF(r), $\mathsf{I_d}[< \times x S\emptyset x] = \mathsf{S}$. But from (AG), $\mathsf{I_h}[\times x S\emptyset] =$ Bill, and $\mathsf{I_h}[x] =$ Hill. And $\langle$Bill, Hill$\rangle \notin \mathsf{I}[<]$, so by SF(r), $\mathsf{I_h}[< \times x S\emptyset x] = \mathsf{N}$. $\mathcal{R}^n t_1 \dots t_n$ is satisfied just in case the n-tuple of the thing assigned to t_1, and ... and the thing assigned to t_n is in the set assigned to the relation symbol. To decide if $\mathcal{R}^n t_1 \dots t_n$ is satisfied, we find out what things are assigned to the term or terms, and then look to see whether the relevant ordered sequence is in the assignment. The simplest sort of case is when there is just one term. $\mathsf{I_d}[\mathcal{R}^1 t] = \mathsf{S}$ just in case $\mathsf{I_d}[t] \in \mathsf{I_d}[\mathcal{R}^1]$. When there is more than one term, we look for the objects taken in order.

SF($\sim$) and SF($\to$) work just as before. And we could work out their consequences on trees or tables for satisfaction as before. In this case, though, to accomodate quantifiers it will be convenient to turn the "trees" on their sides. For this, we begin by constructing the tree in the "forward direction," from left-to-right, and then determine satisfaction the other way — from the branch tips back to the trunk. Where the members of U are $\{\mathsf{m}, \mathsf{n} \ldots\}$, the branch conditions are as follows:

		forward	*backward*
B(s)	$\mathsf{I_d}[\mathcal{S}]$	does not branch	the tip is S iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$
B(r)	$\mathsf{I_d}[\mathcal{R}^n t_1 \ldots t_n]$	branches only for terms	the tip is S iff $\langle \mathsf{I_d}[t_1] \ldots \mathsf{I_d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$
B($\sim$)	$\mathsf{I_d}[\sim\mathcal{P}]$	$\sim$ $\mathsf{I_d}[\mathcal{P}]$	the trunk is S iff the branch is N
B($\to$)	$\mathsf{I_d}[(\mathcal{P} \to \mathcal{Q})]$	$\to$ $\mathsf{I_d}[\mathcal{P}]$ $\mathsf{I_d}[\mathcal{Q}]$	the trunk is S iff the top branch is N or the bottom branch is S (or both)
B($\forall$)	$\mathsf{I_d}[\forall x \mathcal{P}]$	$\forall x$ $\mathsf{I}_{\mathsf{d}(x\vert\mathsf{m})}[\mathcal{P}]$ $\mathsf{I}_{\mathsf{d}(x\vert\mathsf{n})}[\mathcal{P}]$ $\vdots$ one branch for each member of U	The trunk is S iff every branch is S

A formula branches according to its main operator. If it is atomic, it does not branch (or branches only for its terms). (AF) and (AG) are examples of branching for terms, only oriented vertically. If the main operator is $\sim$, a formula has just one branch; if its main operator is $\to$, it has two branches; and if its main operator is $\forall$ it has as many branches as there are members of U. This last condition makes it impractical to construct these trees in all but the most simple cases — and impossible when U is infinite. Still, we can use them to see how the definitions work.

When there are no quantifiers, we should be able to recognize these trees as a mere "sideways" variant of ones we have seen before. Thus, suppose an interpretation L with U = {Bob, Sue, Jim}, $\mathsf{L}[A] = \mathsf{T}$, $\mathsf{L}[B^1] =$ {Sue}, and $\mathsf{L}[C^2] = \{\langle$Bob, Sue$\rangle, \langle$Sue, Jim$\rangle\}$ where variable assignment $\mathsf{d}[x] =$Bob. Then,

$$\text{(AH)}\qquad \begin{array}{cccccc} 1 & & 2 & & 3 & \\ & & \underline{\mathsf{L_d}[\sim A]^{(\mathsf{N})}} & \sim & \underline{\mathsf{L_d}[A]^{(\mathsf{S})}} & \\ \underline{\mathsf{L_d}[\sim A \to Bx]^{(\mathsf{S})}} & \to & & & & \\ & & \underline{\mathsf{L_d}[Bx]^{(\mathsf{N})}} & \vdots\!\text{---} & x^{[\text{Bob}]} & \end{array}$$

The main operator at stage (1) is $\to$; so there are two branches. Bx on the bottom is atomic, so the formula branches no further — though we use TA to calculate the term assignment. On the top at (2), $\sim A$ has main operator $\sim$. So there is one branch. And we are done with the forward part of the tree. Given this, we can calculate satisfaction from the tips, back toward the trunk. Since $\mathsf{L}[A] = \mathsf{T}$, by B(s), the tip at (3) is S. And since this is S, by B($\sim$), the top formula at (2) is N. But since $\mathsf{L_d}[x] = \text{Bob}$, and $\text{Bob} \notin \mathsf{L}[B]$, by B(r), the bottom at (2) is N. And with both the top and bottom at (2) N, by B($\to$), the formula at (1) is S. So $\mathsf{L_d}[\sim A \to Bx] = \mathsf{S}$. You should be able to recognize that the diagram (AH) rotated counterclockwise by 90 degrees would be a mere variant of diagrams we have seen before. And the branch conditions merely implement the corresponding conditions from SF.

Things are more interesting when there are quantifiers. For a quantifier, there are as many branches as there are members of U. Thus working with the same interpretation, consider $\mathsf{L_d}[\forall y{\sim}Cxy]$. If there were just one thing in the universe, say $\mathsf{U} = \{\text{Bob}\}$, the tree would branch as follows,

$$\text{(AI)}\qquad \begin{array}{cccccccc} 1 & & 2 & & 3 & & 4 \\ \underline{\mathsf{L_d}[\forall y{\sim}Cxy]^{(\mathsf{S})}} & \forall y & \underline{\mathsf{L_{d(y|\text{Bob})}}[\sim Cxy]^{(\mathsf{S})}} & \sim & \underline{\mathsf{L_{d(y|\text{Bob})}}[Cxy]^{(\mathsf{N})}} & \vdots\!< & \begin{array}{l} x^{[\text{Bob}]} \\ y^{[\text{Bob}]} \end{array} \end{array}$$

The main operator at (1) is the universal quantifier. Supposing one thing in U, there is the one branch. Notice that the variable assignment d becomes $\mathsf{d}(y|\text{Bob})$. The main operator at (2) is $\sim$. So there is the one branch, carrying forward the assignment $\mathsf{d}(y|\text{Bob})$. The formula at (3) is atomic, so the only branching is for the term assignment. Then, in the backward direction, $\mathsf{L_{d(y|\text{Bob})}}$ still assigns Bob to x; and $\mathsf{L_{d(y|\text{Bob})}}$ assigns Bob to y. Since $\langle \text{Bob, Bob} \rangle \notin \mathsf{L}[C^2]$, the branch at (3) is N; so the branch at (2) is S. And since *all* the branches for the universal quantifier are S, by B($\forall$), the formula at (1) is S.

But L was originally defined with $\mathsf{U} = \{\text{Bob, Sue, Jim}\}$. Thus the quantifier requires not one but three branches, and the proper tree is as follows.

$$
\text{(AJ)}\quad
\begin{array}{cclccc}
1 & & 2 & & 3 & 4\\
 & & \underline{\mathsf{L}_{\mathsf{d}(y|\text{Bob})}[\sim Cxy]^{(\mathsf{S})}} & \sim & \underline{\mathsf{L}_{\mathsf{d}(y|\text{Bob})}[Cxy]^{(\mathsf{N})}} & \vdots\begin{array}{l} x^{[\text{Bob}]}\\ y^{[\text{Bob}]}\end{array}\\
\underline{\mathsf{L}_{\mathsf{d}}[\forall y\sim Cxy]^{(\mathsf{N})}} & \forall y & \underline{\mathsf{L}_{\mathsf{d}(y|\text{Sue})}[\sim Cxy]^{(\mathsf{N})}} & \sim & \underline{\mathsf{L}_{\mathsf{d}(y|\text{Sue})}[Cxy]^{(\mathsf{S})}} & \vdots\begin{array}{l} x^{[\text{Bob}]}\\ y^{[\text{Sue}]}\end{array}\\
 & & \underline{\mathsf{L}_{\mathsf{d}(y|\text{Jim})}[\sim Cxy]^{(\mathsf{S})}} & \sim & \underline{\mathsf{L}_{\mathsf{d}(y|\text{Jim})}[Cxy]^{(\mathsf{N})}} & \vdots\begin{array}{l} x^{[\text{Bob}]}\\ y^{[\text{Jim}]}\end{array}
\end{array}
$$

Now there are three branches for the quantifier. Note the modification of d on each branch, and the way the modified assignments carry forward and are used for evaluation at the tips. $\mathsf{d}(y|\text{Sue})$, say, has the same assignment to x as d, but assigns Sue to y. And similarly for the rest. This time, not all the branches for the universal quantifier are S. So the formula at (1) is N. You should convince yourself that it is S on I_{h} where $\mathsf{h}[x] = $ Jim. It would be S also with the assignment d as above, but formula Cyx.

(AK) on p. 125 is an example for $\forall x[(Sx < x) \to \forall y(Sy + \emptyset) = x)]$ using interpretation I from p. 116 and $\mathscr{L}^{<}_{\text{NT}}$. This case should help you to see how all the parts fit together in a reasonably complex example. It turns out to be helpful to think about the formula in its unabbreviated form, $\forall x(<\!Sxx \to \forall y\!=\!+Sy\emptyset x)$. For this case notice especially how when multiple quantifiers come off, a variable assignment once modified is simply modified again for the new variable. If you follow through the details of this case by the definitions, you are doing well.

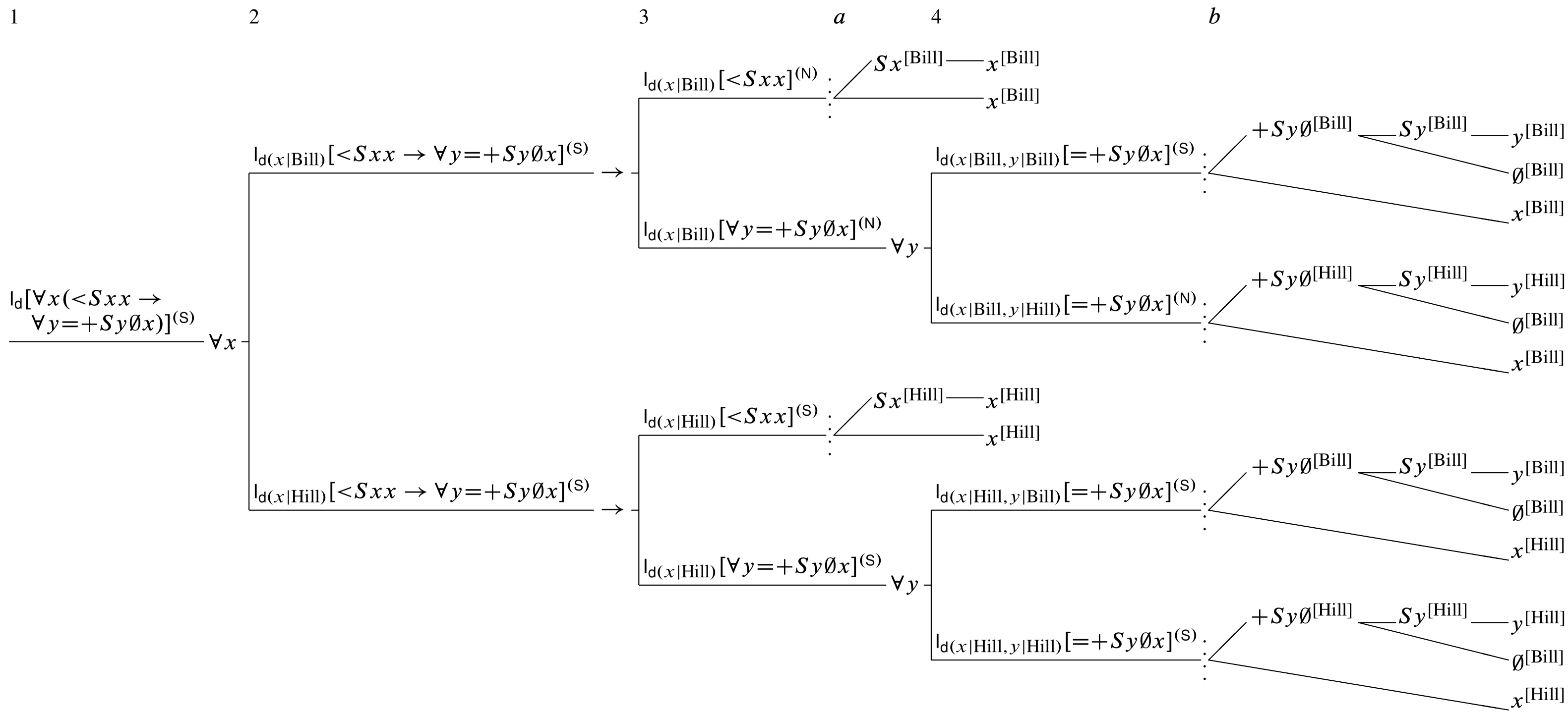
A word of advice: Once you have the idea, constructing these trees to determine satisfaction is a mechanical (and tedious) process. About the only way to go wrong or become confused is by skipping steps or modifying the form of trees. But, very often, skipping steps or modifying form does correlate with confusion! So it is best to stick with the official pattern — and so to follow the way it forces you through definitions SF and TA.

E4.12. Supplement interpretation K for E4.11 so that

$$
\begin{array}{ll}
\mathsf{K} & \mathsf{K}[S] = \mathsf{T}\\
 & \mathsf{K}[H^1] = \{\text{Amy, Bob}\}\\
 & \mathsf{K}[L^2] = \{\langle\text{Amy, Amy}\rangle, \langle\text{Amy, Bob}\rangle, \langle\text{Bob, Bob}\rangle, \langle\text{Bob, Chris}\rangle, \langle\text{Amy, Chris}\rangle\}
\end{array}
$$

Where $\mathsf{d}(x) = $ Amy, $\mathsf{d}(y) = $ Bob, use trees to determine whether the following formulas are satisfied on K with d.

*a. Hx b. Lxa



(AK)

Forward: Since there are two objects in U, there are two branches for each quantifier. At stage (2), for the x-quantifier, d is modified for assignments to x, and at stage (4) for the y-quantifier those assignments are modified again. $<Sxx$ and $=+Sy\emptyset x$ are atomic. Branching for terms continues at stages (a) and (b) in the usual way.

Backward: At the tips for terms apply the variable assignment from the corresponding atomic formula. Thus, in the top at (b) with $\mathsf{d}(x|\mathrm{Bill},\, y|\mathrm{Bill})$, both x and y are assigned to Bill. The assignment to $\emptyset$ comes from I. For (4), recall that $\mathsf{I}[=]$ is automatically $\{\langle \mathrm{Bill}, \mathrm{Bill}\rangle, \langle \mathrm{Hill}, \mathrm{Hill}\rangle\}$. After that, the calculation at each stage is straightforward.

c. Hf^1y

d. $\forall xLyx$

e. $\forall xLxg^2cx$

*f. $\sim\forall x(Hx \to \sim S)$

*g. $\forall y\sim\forall xLxy$

*h. $\forall y\sim\forall xLyx$

i. $\forall x(Hf^1x \to Lxx)$

j. $\forall x(Hx \to \sim\forall y\sim Lyx)$

E4.13. What, if anything, changes with the variable assignment h where $\mathsf{h}[x] =$ Chris and $\mathsf{h}[y] =$ Amy? Challenge: Explain why differences in the initial variable assignment *cannot* matter for the evaluation of (e) - (j).

4.2.4 Truth and Validity

It is a short step from satisfaction to definitions for *truth* and *validity*. Formulas are satisfied or not on an interpretation I together with a variable assignment d. But whether a formula is *true* or *false* on an interpretation depends on satisfaction relative to *every* variable assignment.

TI A formula $\mathscr{P}$ is *true* on an interpretation I iff with any d for I, $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$. $\mathscr{P}$ is *false* on I iff with any d for I, $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$.

A formula is true on I just in case it is satisfied with every variable assignment for I. From (AJ), then, we are already in a position to see that $\forall y\sim Cxy$ is not true on L. For there is a variable assignment d on which it is N. Neither is $\forall y\sim Cxy$ false on L, insofar as it is satisfied when the assignment is h. Since there is an assignment on which it is N, it is not satisfied on *every* assignment, and so is not true. Since there is an assignment on which it is S, it is not N on *every* assignment, and so is not false. In contrast, from (AK), $\forall x[(Sx < x) \to \forall y(Sy + \emptyset) = x)]$ is true on I. For some variable assignment d, the tree shows directly that $\mathsf{I_d}[\forall x[(Sx < x) \to \forall y(Sy + \emptyset) = x)]] = S$. But the reasoning for the tree *makes no assumptions whatsoever about* d. That is, with any variable assignment, we might have reasoned in just the same way, to reach the conclusion that the formula is satisfied. Since it comes out satisfied no matter what the variable assignment may be, by TI, it is true.

In general, if a *sentence* is satisfied on some d for I, then it is satisfied on every d for I. We shall demonstrate this more formally in chapter 8. However, we are already in a position to see the basic idea: In a sentence, every variable is bound; so by the time you get to formulas without quantifiers at the tips of a tree, assignments are of the sort, $\mathsf{d}(x|\mathsf{m}, y|\mathsf{n}\ldots)$ for every variable in the formula; so satisfaction depends just on assignments that are set on the branch itself, and the initial d is irrelevant to

satisfaction at the tips — and thus to evaluation of the formula as a whole. Satisfaction depends on adjustments to the assignment that occur within the tree, rather than on the initial assignment itself. So every starting d has the same result. So if a sentence is satisfied on some d for I, it is satisfied on every d for I, and therefore true on I. Similarly, if a sentence is N on some d for I, it is N on every d for I, and therefore false on I.

In contrast, a formula with free variables may be sensitive to the initial variable assignment. Thus, in the ordinary case, Hx is not true and not false on an interpretation depending on the assignment to x. We have seen this pattern so far in examples and exercises: for formulas with free variables, there may be variable assignments where they are satisfied, and variable assignments where they are not. Therefore the formulas fail to be either true or false by TI. Sentences, on the other hand, are satisfied on every variable assignment if they are satisfied on any, and not satisfied on every assignment if they are not satisfied on any. Therefore the sentences from our examples and exercises come out either true or false.

But a word of caution is in order: Sentences are always true or false on an interpretation. And, in the ordinary case, formulas with free variables are neither true nor false. But this is not always so. $(x = x)$ is true on any I. (Why?) Similarly, $I[Hx] = \mathsf{T}$ if $\mathsf{I}[H] = \mathsf{U}$ and F if $\mathsf{I}[H] = \{\}$. And $\sim\forall x(x = y)$ is true on any I with a U that has more than one member. To see this, suppose for some I, $\mathsf{U} = \{\mathsf{m}, \mathsf{n} \ldots\}$; then for an arbitrary d the tree is as follows,

(AL)

1	2		3	4
			$\mathsf{I}_{\mathsf{d}(x\mid\mathsf{m})}[x = y]$ ⋮	$x^{[\mathsf{m}]}$ $y^{\mathsf{d}[y]}$
			$\mathsf{I}_{\mathsf{d}(x\mid\mathsf{n})}[x = y]$ ⋮	$x^{[\mathsf{n}]}$ $y^{\mathsf{d}[y]}$
$\mathsf{I}_{\mathsf{d}}[\sim\forall x(x = y)]$ $\sim$	$\mathsf{I}_{\mathsf{d}}[\forall x(x = y)]$	$\forall x$	⋮ one branch for each member of U	

No matter what d is like, at most one branch at (3) is S. If $\mathsf{d}[y] = \mathsf{m}$ then the top branch at (3) is S and the rest are N. If $\mathsf{d}[y] = \mathsf{n}$ then the second branch at (3) is S and the others are N. And so forth. So in this case where U has more than one member, at least one branch is N for any d. So the universally quantified expression is N for any d, and the negation at (1) is S for any d. So by TI it is true. So satisfaction for a *formula* may but need not be sensitive to the particular variable assignment under consideration. Again, though, a *sentence* is always true or false depending only on the interpretation. To show that a sentence is true, it is enough to show that it is satisfied on some d, from which it follows that it is satisfied on any. For a formula

with free variables, the matter is more complex — though you can show that such a formula *not* true by finding an assignment that makes it N, and *not* false by finding an assignment that makes it S.

Given the notion of truth, *quantificational validity* works very much as before. Where Γ (Gamma) is a set of formulas, say $\mathsf{I}[\Gamma] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ for each formula $\mathcal{P} \in \Gamma$. Then for any formula $\mathcal{P}$,

QV Γ *quantificationally entails* $\mathcal{P}$ iff there is no quantificational interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$.

Γ quantificationally entails $\mathcal{P}$ when there is no quantificational interpretation that makes the premises true and the conclusion not. If Γ quantificationally entails $\mathcal{P}$ we write, $\Gamma \vDash \mathcal{P}$, and say an argument whose premises are the members of Γ and conclusion is $\mathcal{P}$ is *quantificationally valid*. Γ does not quantificationally entail $\mathcal{P}$ ($\Gamma \nvDash \mathcal{P}$) when there is some quantificational interpretation on which all the premises are true, but the conclusion is not true (notice that there is a difference between being not true, and being false). As before, if $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ are the members of Γ, we sometimes write $\mathcal{Q}_1 \ldots \mathcal{Q}_n \vDash \mathcal{P}$ in place of $\Gamma \vDash \mathcal{P}$. If there are no premises, listing all the members of Γ individually, we simply write $\vDash \mathcal{P}$. If $\vDash \mathcal{P}$, then $\mathcal{P}$ is *logically true*. Notice again the double turnstile $\vDash$, in contrast to the single turnstile $\vdash$ for derivations.

In the quantificational case, *demonstrating* semantic validity is problematic. In the sentential case, we could simply *list* all the ways a sentential interpretation could make basic sentences T or F. In the quantificational case, it is not possible to list all interpretations. Consider just interpretations with universe $\mathbb{N}$: the interpretation of a one-place relation symbol $\mathcal{R}$ might be $\{1\}$ or $\{2\}$ or $\{3\}$ ...; it might be $\{1, 2\}$ or $\{1, 3\}$, or $\{1, 3, 5 \ldots\}$, or whatever. There are infinitely many options for this one relation symbol — and so at least as many for quantificational interpretations in general. Similarly, when the universe is so large, by our methods, we cannot calculate even *satisfaction* and *truth* in arbitrary cases — for quantifiers would have an infinite number of branches. One might begin to suspect that there is no way to demonstrate semantic validity in the quantificational case. There is a way. And we respond to this concern in chapter 7.

For now, though, we rest content with demonstrating *invalidity*. To show that an argument is invalid, we do not need to consider all possible interpretations; it is enough to find one interpretation on which the premises are true, and the conclusion is not. (Compare the invalidity format from chapter 1 and "shortcut" truth tables in this chapter.) An argument is quantificationally valid just in case there is no I

on which its premises are true, and its conclusion is not true. So to show that an argument is not quantificationally valid, it is sufficient to produce an interpretation that violates this condition — an interpretation on which its premises are true and conclusion is not. This should be enough at least to let us see *how* the definitions work, and we postpone the larger question about showing quantificational validity to later.

For now, then, our idea is to produce an interpretation, and then to use trees in order to show that the interpretation makes premises true, but the conclusion not. Thus, for example, for $\mathcal{L}_q$ we can show that $\sim\forall xPx \nvDash \sim Pa$ — that an argument with premise $\sim\forall xPx$ and conclusion $\sim Pa$ is not quantificationally valid. To see this, consider an I with $\mathsf{U} = \{1, 2\}$, $\mathsf{I}[P] = \{1\}$, and $\mathsf{I}[a] = 1$. Then $\sim\forall xPx$ is T on I.

$$
\begin{array}{lllll}
 & 1 & 2 & 3 & 4\\
 & & & \underline{\mathsf{I}_{\mathsf{d}(x|1)}[Px]^{(\mathsf{S})}} & \underline{\quad}\; x^{[1]}\\
\text{(AM)} & \underline{\mathsf{I}_{\mathsf{d}}[\sim\forall xPx]^{(\mathsf{S})}}\;\sim & \underline{\mathsf{I}_{\mathsf{d}}[\forall xPx]^{(\mathsf{N})}}\;\forall x & & \\
 & & & \underline{\mathsf{I}_{\mathsf{d}(x|2)}[Px]^{(\mathsf{N})}} & \underline{\quad}\; x^{[2]}
\end{array}
$$

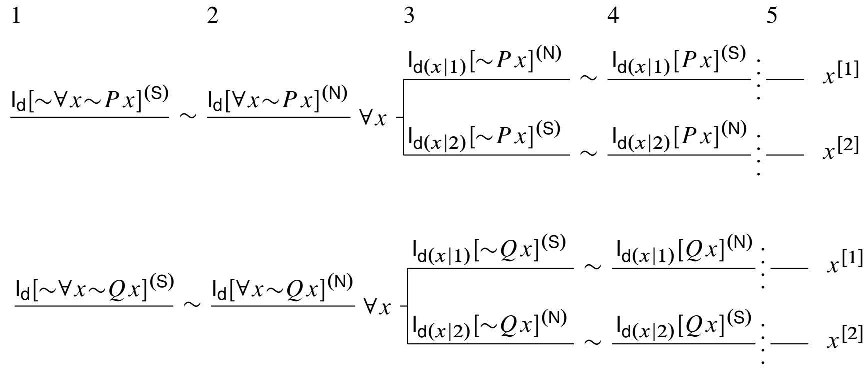
$\sim\forall xPx$ is satisfied with this d for I; since it is a sentence it is satisfied with any d for I. So by TI it is true on I. But $\sim Pa$ is not true on this I.

$$
\begin{array}{lll}
1 & 2 & 3\\
\underline{\mathsf{I}_{\mathsf{d}}[\sim Pa]^{(\mathsf{N})}}\;\sim & \underline{\mathsf{I}_{\mathsf{d}}[Pa]^{(\mathsf{S})}} & \underline{\quad}\; a^{[1]}
\end{array}
$$

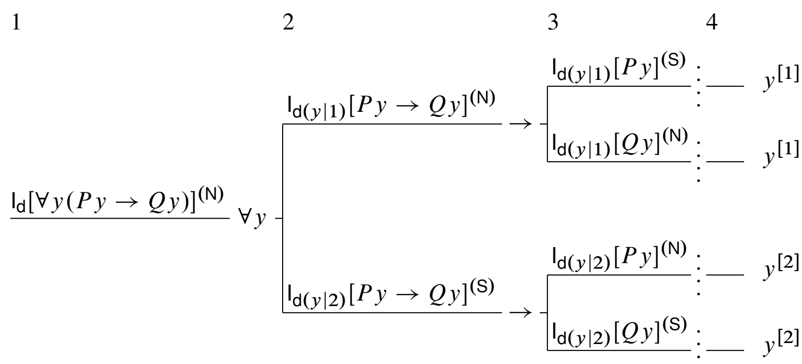
By TA(c), $\mathsf{I}_{\mathsf{d}}[a] = \mathsf{I}[a]$. So the assignment to a is 1 and the formula at (2) is satisfied, so that the formula at (1) is not. So by TI, $\mathsf{I}[\sim Pa] \neq \mathsf{T}$. So there is an interpretation on which the premise is true and the conclusion is not; so $\sim\forall xPx \nvDash \sim Pa$, and the argument is not quantificationally valid. Notice that it is sufficient to show that the conclusion is not true — which is not always the same as showing that the conclusion is false.

Here is another example. We show that $\sim\forall x\sim Px$, $\sim\forall x\sim Qx \nvDash \forall y(Py \to Qy)$. One way to do this is with an I that has $\mathsf{U} = \{1, 2\}$ where $\mathsf{I}[P] = \{1\}$ and $\mathsf{I}[Q] = \{2\}$. Then the premises are true.

(AN)



To make $\sim\forall x{\sim}Px$ true, we require that there is at least one thing in $\mathsf{I}[P]$. We accomplish this by putting 1 in its interpretation. This makes the top branch at stage (4) S; this makes the top branch at (3) N; so the quantifier at (2) is N and the formula at (1) comes out S. Since it is a sentence and satisfied on the arbitrary assignment, it is true. $\sim\forall x{\sim}Qx$ is true for related reasons. For it to be true, we require at least one thing in $\mathsf{I}[Q]$. This is accomplished by putting 2 in its interpretation. But this interpretation does not make the conclusion true.



The conclusion is not satisfied so long as something is in $\mathsf{I}[P]$ but not in $\mathsf{I}[Q]$. We accomplish this by making the thing in the interpretation of P *different* from the thing in the interpretation of Q. Since 1 is in $\mathsf{I}[P]$ but not in $\mathsf{I}[Q]$, there is an S/N pair at (4), so that the top branch at (2) is N and the formula at (1) is N. Since the formula is not satisfied, by TI it is not true. And since there is an interpretation on which the premises are true and the conclusion is not, by QV, the argument is not quantificationally valid.

In general, to show that an argument is not quantificationally valid, you want to think "backward" to see what kind of interpretation you need to make the premises

true but the conclusion not true. It is to your advantage to think of simple interpretations. Remember that U need only be non-empty. So it will often do to work with universes that have just one or two members. And the interpretation of a relation symbol might even be empty! It is often convenient to let the universe be some set of integers. And, if there is any interpretation that demonstrates invalidity, there is sure to be one whose universe is some set of integers — but we will get to this in Part III.

E4.14. For language $\mathcal{L}_q$ consider an interpretation I such that $\mathsf{U} = \{1, 2\}$, and

$$\mathsf{I} \quad \begin{aligned} &\mathsf{I}[a] = 1 \\ &\mathsf{I}[A] = \mathsf{T} \\ &\mathsf{I}[P^1] = \{1\} \\ &\mathsf{I}[f^1] = \{\langle 1, 2\rangle, \langle 2, 1\rangle\} \end{aligned}$$

Use interpretation I and trees to show that (a) below is not quantificationally valid. Then each of the others can be shown to be invalid on an interpretation I* that modifies just one of the main parts of I. Produce the modified interpretations, and use them to show that the other arguments also are invalid. Hint: If you are having trouble finding the appropriate modified interpretation, try working out the trees on I, and think about what changes to the interpretation would have the results you want.

a. $Pa \not\models \forall x Px$

b. $\sim Pa \not\models \forall x {\sim} Px$

*c. $\forall x Pf^1x \not\models \forall x Px$

d. $\forall x(Px \rightarrow Pf^1x) \not\models \forall x(Pf^1x \rightarrow Px)$

e. $\forall x Px \rightarrow A \not\models \forall x(Px \rightarrow A)$

E4.15. Find interpretations and use trees to demonstrate each of the following. Be sure to explain why your interpretations and trees have the desired result.

*a. $\forall x(Qx \rightarrow Px) \not\models \forall x(Px \rightarrow Qx)$

b. $\forall x(Px \rightarrow Qx), \forall x(Rx \rightarrow {\sim} Px) \not\models \forall y(Ry \rightarrow Qy)$

*c. ${\sim}\forall x Px \not\models {\sim} Pa$

d. ${\sim}\forall x Px \not\models \forall x {\sim} Px$

e. $\forall xPx \to \forall xQx,\ Qb \not\models Pa \to \forall xQx$

f. $\sim(A \to \forall xPx) \not\models \forall x(A \to \sim Px)$

g. $\forall x(Px \to Qx), \sim Qa \not\models \forall x\sim Px$

*h. $\sim\forall y\forall xRxy \not\models \forall x\sim\forall yRxy$

i. $\forall x\forall y(Rxy \to Ryx), \forall x\sim\forall y\sim Rxy \not\models \forall xRxx$

j. $\forall x\forall y[y = f^1x \to \sim(x = f^1y)] \not\models \forall x(Px \to Pf^1x)$

4.2.5 Abbreviations

Finally, we turn to applications for abbreviations. Consider first a tree for $(\mathcal{P} \wedge \mathcal{Q})$, that is for $\sim(\mathcal{P} \to \sim\mathcal{Q})$.

$$\begin{array}{lllll} & 1 & 2 & 3 & 4 \\ \text{(AO)} & \underline{\mathsf{I_d}[\sim(\mathcal{P}\to\sim\mathcal{Q})]}\ \sim & \underline{\mathsf{I_d}[\mathcal{P}\to\sim\mathcal{Q}]}\ \to & \begin{array}{|l}\underline{\mathsf{I_d}[\mathcal{P}]}\\ \\ \underline{\mathsf{I_d}[\sim\mathcal{Q}]}\end{array}\ \sim & \underline{\mathsf{I_d}[\mathcal{Q}]}\end{array}$$

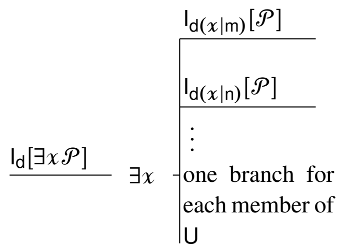
The formula at (1) is satisfied iff the formula at (2) is not. But the formula at (2) is not satisfied iff the top at (3) is satisfied and the bottom is not satisfied. And the bottom at (3) is not satisfied iff the formula at (4) is satisfied. So the formula at (1) is satisfied iff $\mathcal{P}$ is satisfied and $\mathcal{Q}$ is satisfied. The only way for $(\mathcal{P} \wedge \mathcal{Q})$ to be satisfied on some I and d, is for $\mathcal{P}$ and $\mathcal{Q}$ both to be satisfied on that I and d. If either $\mathcal{P}$ or $\mathcal{Q}$ is not satisfied, then $(\mathcal{P} \wedge \mathcal{Q})$ is not satisfied. Reasoning similarly for $\vee$, $\leftrightarrow$, and $\exists$, we get the following derived branch conditions.

$$\text{B}(\wedge)\quad \underline{\mathsf{I_d}[(\mathcal{P}\wedge\mathcal{Q})]}\ \wedge \begin{array}{|l}\underline{\mathsf{I_d}[\mathcal{P}]}\\ \\ \underline{\mathsf{I_d}[\mathcal{Q}]}\end{array}\qquad \text{the trunk is } \mathsf{S} \text{ iff both branches are } \mathsf{S}$$

$$\text{B}(\vee)\quad \underline{\mathsf{I_d}[(\mathcal{P}\vee\mathcal{Q})]}\ \vee \begin{array}{|l}\underline{\mathsf{I_d}[\mathcal{P}]}\\ \\ \underline{\mathsf{I_d}[\mathcal{Q}]}\end{array}\qquad \text{the trunk is } \mathsf{S} \text{ iff at least one branch is } \mathsf{S}$$

$$\text{B}(\leftrightarrow)\quad \underline{\mathsf{I_d}[(\mathcal{P}\leftrightarrow\mathcal{Q})]}\leftrightarrow \begin{array}{|l}\underline{\mathsf{I_d}[\mathcal{P}]}\\ \\ \underline{\mathsf{I_d}[\mathcal{Q}]}\end{array}\qquad \text{the trunk is } \mathsf{S} \text{ iff both branches are } \mathsf{S} \text{ or both are } \mathsf{N}$$

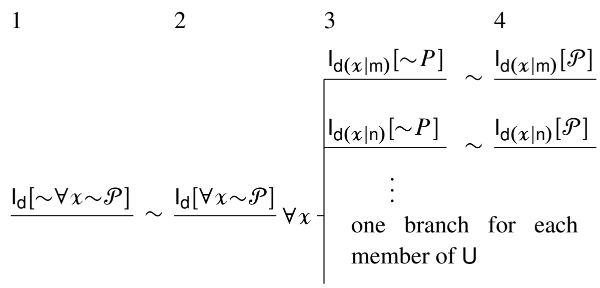
B(∃)



The trunk is S iff at least one branch is S

The cases for $\wedge$, $\vee$, and $\leftrightarrow$ work just as in the sentential case. For the last, consider a tree for $\sim\forall x\sim\mathcal{P}$, that is for $\exists x\mathcal{P}$.

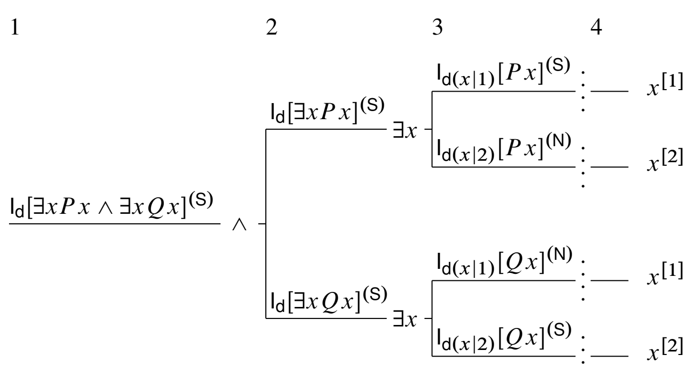
(AP)



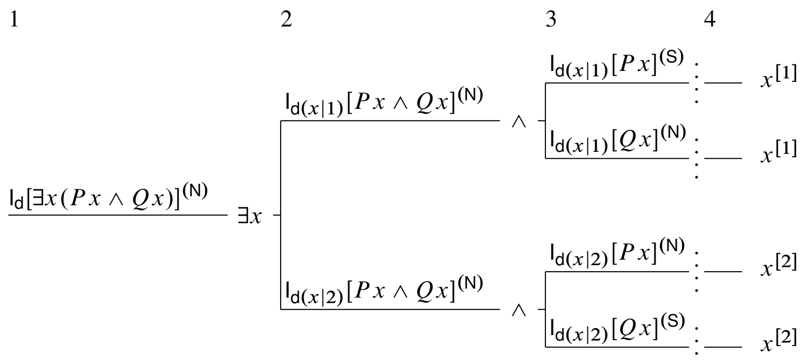
The formula at (1) is satisfied iff the formula at (2) is not. But the formula at (2) is not satisfied iff at least one of the branches at (3) is not satisfied. And for a branch at (3) to be not satisfied, the corresponding branch at (4) has to be satisfied. So $\exists x\mathcal{P}$ is satisfied on I with assignment d iff for some $\mathsf{o} \in \mathsf{U}$, $\mathcal{P}$ is satisfied on I with $\mathsf{d}[x|\mathsf{o}]$; if there is no such $\mathsf{o} \in \mathsf{U}$, then $\exists x\mathcal{P}$ is N on I with d.

Given derived branch conditions, we can work directly with abbreviations in trees for determining satisfaction and truth. And the definition of validity applies in the usual way. Thus, for example, $\exists xPx \wedge \exists xQx \not\vDash \exists x(Px \wedge Qx)$. To see this, consider an I with $\mathsf{I} = \{1, 2\}$, $\mathsf{I}[P] = \{1\}$ and $\mathsf{I}[Q] = \{2\}$. The premise, $\exists xPx \wedge \exists xQx$ is true on I. To see this, we construct a tree, making use of derived clauses as necessary.

(AQ)



The existentials are satisfied because at least one branch is satisfied, and the conjunction because both branches are satisfied, according to derived conditions B($\exists$) and B($\wedge$). So the formula is satisfied, and because it is a sentence, is true. But the conclusion, $\exists x(Px \wedge Qx)$ is not true.



The conjunctions at (2) are not satisfied, in each case because not both branches at (3) are satisfied. And the existential at (1) requires that at least one branch at (2) be satisfied; since none is satisfied, the main formula $\exists x(Px \wedge Qx)$ is not satisfied, and so by TI not true. Since there is an interpretation on which the premise is true and the conclusion is not, by QV, $\exists xPx \wedge \exists xQx \nvDash \exists x(Px \wedge Qx)$. As we will see in the next chapter, the intuitive point is simple: just because something is P and something is Q, it does not follow that something is both P and Q. And this is just what our interpretation I illustrates.

E4.16. On p. 132 we say that reasoning similar to that for $\wedge$ results in other branch conditions. Give the reasoning similar to that for $\wedge$ and $\exists$ to demonstrate from trees the conditions B($\vee$) and B($\leftrightarrow$).

E4.17. Produce interpretations to demonstrate each of the following. Use trees, with derived clauses as necessary, to demonstrate your results. Be sure to explain why your interpretations and trees have the results they do.

*a. $\exists xPx \nvDash \forall yPy$

b. $\exists xPx \nvDash \exists y(Py \wedge Qy)$

c. $\exists xPx \nvDash \exists yPf^1y$

d. $Pa \to \forall xQx \nvDash \exists xPx \to \forall xQx$

Semantics Quick Reference (quantificational)

For a quantificational language $\mathscr{L}$, a *quantificational interpretation* I consists of a nonempty set U, the *universe* of the interpretation, along with,

QI (s) An assignment of a truth value $\mathsf{I}[\mathscr{S}]$ to each sentence letter $\mathscr{S}$ of $\mathscr{L}$.

(c) An assignment of a member $\mathsf{I}[c]$ of U to each constant symbol c of $\mathscr{L}$.

(r) An assignment of an n-place relation $\mathsf{I}[\mathscr{R}^n]$ on U to each n-place relation symbol $\mathscr{R}^n$ of $\mathscr{L}$, where $\mathsf{I}[=]$ is always assigned $\{\langle \mathsf{o}, \mathsf{o}\rangle \mid \mathsf{o} \in \mathsf{U}\}$.

(f) An assignment of a total n-place function $\mathsf{I}[\mathscr{h}^n]$ from U^n to U, to each n-place function symbol $\mathscr{h}^n$ of $\mathscr{L}$.

Given a language $\mathscr{L}$ and interpretation I, a *variable assignment* d is a total function from the variables of $\mathscr{L}$ to objects in the universe U. Then for any interpretation I, variable assignment d, and term t,

TA (c) If c is a constant, then $\mathsf{I_d}[c] = \mathsf{I}[c]$.

(v) If x is a variable, then $\mathsf{I_d}[x] = \mathsf{d}[x]$.

(f) If $\mathscr{h}^n$ is a function symbol and $t_1 \dots t_n$ are terms, then $\mathsf{I_d}[\mathscr{h}^n t_1 \dots t_n] = \mathsf{I}[\mathscr{h}^n]\langle \mathsf{I_d}[t_1] \dots \mathsf{I_d}[t_n]\rangle$.

For any interpretation I with variable assignment d,

SF (s) If $\mathscr{S}$ is a sentence letter, then $\mathsf{I_d}[\mathscr{S}] = \mathsf{S}$ iff $\mathsf{I}[\mathscr{S}] = \mathsf{T}$; otherwise $\mathsf{I_d}[\mathscr{S}] = \mathsf{N}$.

(r) If $\mathscr{R}^n$ is an n-place relation symbol and $t_1 \dots t_n$ are terms, then $\mathsf{I_d}[\mathscr{R}^n t_1 \dots t_n] = \mathsf{S}$ iff $\langle \mathsf{I_d}[t_1] \dots \mathsf{I_d}[t_n]\rangle \in \mathsf{I}[\mathscr{R}^n]$; otherwise $\mathsf{I_d}[\mathscr{R}^n t_1 \dots t_n] = \mathsf{N}$.

($\sim$) If $\mathscr{P}$ is a formula, then $\mathsf{I_d}[\sim\mathscr{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$; otherwise $\mathsf{I_d}[\sim\mathscr{P}] = \mathsf{N}$.

($\rightarrow$) If $\mathscr{P}$ and $\mathscr{Q}$ are formulas, then $\mathsf{I_d}[(\mathscr{P} \rightarrow \mathscr{Q})] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$ or $\mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$ (or both); otherwise $\mathsf{I_d}[(\mathscr{P} \rightarrow \mathscr{Q})] = \mathsf{N}$.

($\forall$) If $\mathscr{P}$ is a formula and x is a variable, then $\mathsf{I_d}[\forall x\mathscr{P}] = \mathsf{S}$ iff for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}] = \mathsf{S}$; otherwise $\mathsf{I_d}[\forall x\mathscr{P}] = \mathsf{N}$.

SF′ ($\wedge$) If $\mathscr{P}$ and $\mathscr{Q}$ are formulas, then $\mathsf{I_d}[(\mathscr{P} \wedge \mathscr{Q})] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$ and $\mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$; otherwise $\mathsf{I_d}[(\mathscr{P} \wedge \mathscr{Q})] = \mathsf{N}$.

($\vee$) If $\mathscr{P}$ and $\mathscr{Q}$ are formulas, then $\mathsf{I_d}[(\mathscr{P} \vee \mathscr{Q})] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$ or $\mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$ (or both); otherwise $\mathsf{I_d}[(\mathscr{P} \vee \mathscr{Q})] = \mathsf{N}$.

($\leftrightarrow$) If $\mathscr{P}$ and $\mathscr{Q}$ are formulas, then $\mathsf{I_d}[(\mathscr{P} \leftrightarrow \mathscr{Q})] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{I_d}[\mathscr{Q}]$; otherwise $\mathsf{I_d}[(\mathscr{P} \leftrightarrow \mathscr{Q})] = \mathsf{N}$.

($\exists$) If If $\mathscr{P}$ is a formula and x is a variable, then $\mathsf{I_d}[\exists x\mathscr{P}] = \mathsf{S}$ iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}] = \mathsf{S}$; otherwise $\mathsf{I_d}[\exists x\mathscr{P}] = \mathsf{N}$.

TI A formula $\mathscr{P}$ is *true* on an interpretation I iff with any d for I, $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$. $\mathscr{P}$ is *false* on I iff with any d for I, $\mathsf{I_d}[\mathscr{P}] = \mathsf{N}$.

QV Γ *quantificationally entails* $\mathscr{P}$ ($\Gamma \vDash \mathscr{P}$) iff there is no quantificational interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$.

If $\Gamma \vDash \mathscr{P}$, an argument whose premises are the members of Γ and conclusion is $\mathscr{P}$ is *quantificationally valid*.

e. $\forall x\exists yRxy \nvDash \exists y\forall xRxy$

f. $\forall xPx \leftrightarrow \forall xQx, \exists x\exists y(Px \wedge Qy) \nvDash \exists y(Py \leftrightarrow Qy)$

*g. $\forall x(\exists yRxy \leftrightarrow \sim A) \nvDash \exists xRxx \vee A$

h. $\exists x(Px \wedge \exists yQy) \nvDash \exists x\forall y(Px \wedge Qy)$

i. $\forall x\forall y(Px \vee Qxy), \exists xPx \nvDash \exists x\exists yQxy$

j. $\exists x\exists y\sim(x = y) \nvDash \forall x\forall y\exists z[\sim(x = z) \wedge \sim(y = z)]$

E4.18. Produce an interpretation to demonstrate each of the following (now in $\mathscr{L}_{NT}^{<}$). Use trees to demonstrate your results. Be sure to explain why your interpretations and trees have the results they do. Hint: When there are no premises, all you need is an interpretation where the expression is not true. You need not use the *standard* interpretation! In some cases, it may be convenient to produce only that part of the tree which is necessary for the result.

a. $\nvDash \forall x(x < Sx)$

b. $\nvDash (S\emptyset + S\emptyset) = SS\emptyset$

c. $\nvDash \exists x\sim[(x \times x) = x]$

*d. $\nvDash \forall x\forall y[\sim(x = y) \rightarrow (x < y \vee y < x)]$

e. $\nvDash \forall x\forall y\forall z[(x < y \wedge y < z) \rightarrow x < z]$

E4.19. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Quantificational interpretations.

b. Term assignments, satisfaction and truth.

c. Quantificational validity.

Chapter 5

Translation

We have introduced logical validity from chapter 1, along with notions of semantic validity from chapter 4, and validity in an axiomatic derivation system from chapter 3. But logical validity applies to arguments expressed in ordinary language, where the other notions apply to arguments expressed in a formal language. Our guiding idea has been to *use* the formal notions with application to ordinary arguments via *translation* from ordinary language to the formal ones. It is to the translation task that we now turn. After some general discussion, we will take up issues specific to the sentential, and then the quantificational, cases.

5.1 General

As speakers of ordinary languages (at least English for those reading this book) we presumably have some understanding of the conditions under which ordinary language sentences are true and false. Similarly, we now have an understanding of the conditions under which sentences of our formal languages are true and false. This puts us in a position to recognize when the conditions under which ordinary sentences are true are the *same* as the conditions under which formal sentences are true. And that is what we want: Our goal is to translate the premises and conclusion of ordinary arguments into formal expressions that are true when the ordinary sentences are true, and false when the ordinary sentences are false. Insofar as validity has to do with conditions under which sentences are true and false, our translations should thus be an adequate basis for evaluations of validity.

We can put this point with greater precision. Formal sentences are true and false relative to interpretations. As we have seen, many different interpretations of a formal language are possible. In the sentential case, any sentence letter can be true or false

— so that there are 2^n ways to interpret any n sentence letters. When we specify an interpretation, we select just one of the many available options. Thus, for example, we might set $\mathsf{I}[B] = \mathsf{T}$ and $\mathsf{I}[H] = \mathsf{F}$. But we might also specify an interpretation as follows,

$$\text{(A)} \qquad \begin{array}{l} B\text{: Bill is happy} \\ H\text{: Hillary is happy} \end{array}$$

intending B to take the same truth value as 'Bill is happy' and H the same as 'Hillary is happy'. In this case, the single specification might result in different interpretations, depending on how the world is: Depending on how Bill and Hillary are, the interpretation of B might be true or false, and similarly for H. That is, specification (A) is really a *function* from ways the world could be (from complete and consistent stories) to interpretations of the sentence letters. It results in a specific or *intended* interpretation relative to any way the world could be. Thus, where ω (omega) ranges over ways the world could be, (A) is a function II which results in an intended interpretation II_ω corresponding to any such way — thus $\mathsf{II}_\omega[B]$ is T if Bill is happy at ω and F if he is not.

When we set out to translate some ordinary sentences into a formal language, we always begin by specifying an intended interpretation of the formal language for arbitrary ways the world can be. In the sentential case, this typically takes the form of a specification like (A). Then for any way the world can be ω there is an intended interpretation II_ω of the formal language. Given this, for an ordinary sentence $\mathcal{A}$, the aim is to produce a formal counterpart $\mathcal{A}'$ such that $\mathsf{II}_\omega[\mathcal{A}'] = \mathsf{T}$ iff the ordinary $\mathcal{A}$ is true in world ω. This is the content of saying we want to produce formal expressions that "are true when the ordinary sentences are true, and false when the ordinary sentences are false." In fact, we can turn this into a *criterion of goodness* for translation.

CG Given some ordinary sentence $\mathcal{A}$, a translation consisting of an interpretation function II and formal sentence $\mathcal{A}'$ is *good* iff it captures available sentential/quantificational structure and, where ω is any way the world can be, $\mathsf{II}_\omega[\mathcal{A}'] = \mathsf{T}$ iff $\mathcal{A}$ is true at ω.

If there is a collection of sentences, a translation is *good* given an II where *each* member $\mathcal{A}$ of the collection of sentences has an $\mathcal{A}'$ such that $\mathsf{II}_\omega[\mathcal{A}'] = \mathsf{T}$ iff $\mathcal{A}$ is true at ω. Set aside the question of what it is to capture "available" sentential/quantificational structure, this will emerge as we proceed. For now, the point is simply that we want formal sentences to be true on intended interpretations when originals are true at

corresponding worlds, and false on intended interpretations when originals are false. CG says that this correspondence is necessary for goodness. And, supposing that sufficient structure is reflected, according to CG such correspondence is sufficient as well.

The situation might be pictured as follows. There is a specification II which results in an intended interpretation corresponding to any way the world can be. And corresponding to ordinary sentences $\mathcal{P}$ and $\mathcal{Q}$ there are formal sentences $\mathcal{P}'$ and $\mathcal{Q}'$. Then,

$$
\begin{array}{cccc}
\boxed{\begin{array}{c}\mathsf{II}_{\omega_1}[\mathcal{P}'] = \mathsf{T}\\ \mathsf{II}_{\omega_1}[\mathcal{Q}'] = \mathsf{T}\end{array}} &
\boxed{\begin{array}{c}\mathsf{II}_{\omega_2}[\mathcal{P}'] = \mathsf{T}\\ \mathsf{II}_{\omega_2}[\mathcal{Q}'] = \mathsf{F}\end{array}} &
\boxed{\begin{array}{c}\mathsf{II}_{\omega_3}[\mathcal{P}'] = \mathsf{F}\\ \mathsf{II}_{\omega_3}[\mathcal{Q}'] = \mathsf{T}\end{array}} &
\boxed{\begin{array}{c}\mathsf{II}_{\omega_4}[\mathcal{P}'] = \mathsf{F}\\ \mathsf{II}_{\omega_4}[\mathcal{Q}'] = \mathsf{F}\end{array}} \\
\mathsf{II} \Uparrow & \mathsf{II} \Uparrow & \mathsf{II} \Uparrow & \mathsf{II} \Uparrow \\
\boxed{\omega_1 \begin{array}{c}\mathcal{P}\colon \textit{true}\\ \mathcal{Q}\colon \textit{true}\end{array}} &
\boxed{\omega_2 \begin{array}{c}\mathcal{P}\colon \textit{true}\\ \mathcal{Q}\colon \textit{false}\end{array}} &
\boxed{\omega_3 \begin{array}{c}\mathcal{P}\colon \textit{false}\\ \mathcal{Q}\colon \textit{true}\end{array}} &
\boxed{\omega_4 \begin{array}{c}\mathcal{P}\colon \textit{false}\\ \mathcal{Q}\colon \textit{false}\end{array}}
\end{array}
$$

The interpretation function results in an intended interpretation corresponding to each world. The translation is good only if no matter how the world is, the values of $\mathcal{P}'$ and $\mathcal{Q}'$ on the intended interpretations match the values of $\mathcal{P}$ and $\mathcal{Q}$ at the corresponding worlds or stories.

The premises and conclusion of an argument are some sentences. So the translation of an argument is *good* iff the translation of the sentences that are its premises and conclusion is good. And good translations of arguments put us in a position to *use* our machinery to evaluate questions of validity. Of course, so far, this is an abstract description of what we are about to do. But it should give some orientation, and help you understand what is accomplished as we proceed.

5.2 Sentential

We begin with the sentential case. Again, the general idea is to *recognize* when the conditions under which ordinary sentences are true are the *same* as the conditions under which formal ones are true. Surprisingly perhaps, the hardest part is on the side of recognizing truth conditions in ordinary language. With this in mind, let us begin with some definitions whose application is to expressions of *ordinary* language; after that, we will turn to a procedure for translation, and to discussion of particular operators.

5.2.1 Some Definitions

In this section, we introduce a series of definitions whose application is to ordinary language. These definitions are not meant to compete with anything you have learned in English class. They are rather specific to our purposes. With the definitions under our belt, we will be able to say with some precision what we want to do.

First, a *declarative sentence* is a sentence which has a truth value. ‘Snow is white’ and ‘Snow is green’ are declarative sentences — the first true and the second false. ‘Study harder!’ and ‘Why study?’ are sentences, but not declarative sentences. Given this, a *sentential operator* is an expression containing “blanks” such that when the blanks are filled with declarative sentences, the result is a declarative sentence. In ordinary speech and writing, such blanks do not typically appear (!) however punctuation and expression typically fill the same role. Examples are,

> John believes that ____
>
> John heard that ____
>
> It is not the case that ____
>
> ____ and ____

‘John believes that snow is white’, ‘John believes that snow is green’, and ‘John believes that dogs fly’ are all sentences — some more plausibly true than others. Still, ‘Snow is white’, ‘Snow is green’, and ‘Dogs fly’ are all declarative sentences, and when we put them in the blank of ‘John believes that ____’ the result is a declarative sentence, where the same would be so for any declarative sentence in the blank; so ‘John believes that ____’ is a sentential operator. Similarly, ‘Snow is white and dogs fly’ is a declarative sentence — a false one, since dogs do not fly. And, so long as we put declarative sentences in the blanks of ‘____ and ____’ the result is always a declarative sentence. So ‘____ and ____’ is a sentential operator. In contrast,

> When ____
>
> ____ is white ____

are not sentential operators. Though ‘Snow is white’ is a declarative sentence, ‘When snow is white’ is an adverbial clause, not a declarative sentence. And, though ‘Dogs fly’ and ‘Snow is green’ are declarative sentences, ‘Dogs fly is white snow is green’ is ungrammatical nonsense. If you can think of even one case where putting declarative

sentences in the blanks of an expression does not result in a declarative sentence, then the expression is not a sentential operator. So these are not sentential operators.

Now, as in these examples, we can think of some declarative sentences as generated by the combination of sentential operators with other declarative sentences. Declarative sentences generated from other sentences by means of sentential operators are *compound*; all others are *simple*. Thus, for example, 'Bob likes Mary' and 'Socrates is wise' are simple sentences, they do not have a declarative sentence in the blank of any operator. In contrast, 'John believes that Bob likes Mary' and 'Jim heard that John believes that Bob likes Mary' are compound. The first has a simple sentence in the blank of 'John believes that ____'. The second puts a compound in the blank of 'Jim heard that ____'.

For cases like these, the *main operator* of a compound sentence is that operator not in the blank of any other operator. The main operator of 'John believes that Bob likes Mary' is 'John believes that ____'. And the main operator of 'Jim heard that John believes that Bob likes Mary' is 'Jim heard that ____'. The main operator of 'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob' is '____ and ____', for that is the operator not in the blank of any other. Notice that the main operator of a sentence need not be the *first* operator in the sentence. Observe also that operator structure may not be obvious. Thus, for example, 'Jim heard that Bob likes Sue and Sue likes Jim' is capable of different interpretations. It might be, 'Jim heard that Bob likes Sue and Sue likes Jim' with main operator, 'Jim heard that ____' and the compound, 'Bob likes Sue and Sue likes Jim' in its blank. But it might be 'Jim heard that Bob likes Sue and Sue likes Jim' with main operator, '____ and ____'. The question is what Jim heard, and what the 'and' joins. As suggested above, punctuation and expression often serve in ordinary language to disambiguate confusing cases. These questions of interpretation are not peculiar to our purposes! Rather they are the ordinary questions that might be asked about what one is saying. The underline structure serves to disambiguate claims, to make it very clear how the operators apply.

When faced with a compound sentence, the best approach is start with the whole, rather than the parts. So begin with blank(s) for the main operator. Thus, as we have seen, the main operator of 'It is not the case that Bob likes Sue, and it is not the case that Sue likes Bob' is '____ and ____'. So begin with lines for that operator, 'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob' (leaving space for lines above). Now focus on the sentence in one of the blanks, say the left; that sentence, 'It is not the case that Bob likes Sue' is is a compound with main operator, 'it is not the case that ____'. So add the underline for that operator, 'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob'. The sentence in the blank

of 'it is not the case that ____' is simple. So turn to the sentence in the right blank of the main operator. That sentence has main operator 'it is not the case that ____'. So add an underline. In this way we end up with, 'It is not the case that Bob likes Sue and it is not the case that Sue likes Bob' where, again, the sentence in the last blank is simple. Thus, a complex problem is reduced to ones that are progressively more simple. Perhaps this problem was obvious from the start. But this approach will serve you well as problems get more complex!

We come finally to the key notion of a *truth functional* operator. A sentential operator is *truth functional* iff any compound generated by it has its truth value wholly determined by the truth values of the sentences in its blanks. We will say that the truth value of a compound is "determined" by the truth values of sentences in blanks just in case there is no way to switch the truth value of the whole while keeping truth values of sentences in the blanks constant. This leads to a test for truth functionality: We show that an operator is *not* truth functional, if we come up with some situation(s) where truth values of sentences in the blanks are the same, but the truth value of the resulting compounds are not. To take a simple case, consider 'John believes that ____'. If things are pretty much as in the actual world, 'Dogs fly' and 'There is a Santa' are both false. But if John is a small child it may be that,

		Dogs fly
(B)	John believes that	There is a Santa
	F/T	***F***

the compound is false with one in the blank, and true with the other. Thus the truth value of the compound is not wholly determined by the truth value of the sentence in the blank. We have found a situation where sentences with the same truth value in the blank result in a different truth value for the whole. Thus 'John believes that ____' is not truth functional. We might make the same point with a pair of sentences that are true, say 'Dogs bark' and 'There are infinitely many prime numbers' (be clear in your mind about how this works).

As a second example, consider, '____ because ____'. Suppose 'You are happy', 'You got a good grade', 'There are fish in the sea' and 'You woke up this morning' are all true.

	You are happy		You got a good grade
(C)	There are fish in the sea	because	You work up this morning
	T	***T/F***	***T***

Still, it is natural to think that, the truth value of the compound, 'You are happy because you got a good grade' is true, but 'There are fish in the sea because you woke up this morning' is false. For perhaps getting a good grade makes you happy, but the fish in the sea have nothing to do with your waking up. Thus there are consistent

situations or stories where sentences in the blanks have the same truth values, but the compounds do not. Thus, by the definition, '____ because ____' is not a truth functional operator. To show that an operator is not truth functional it is sufficient to produce some situation of this sort: where truth values for sentences in the blanks match, but truth values for the compounds do not. Observe that sentences in the blanks are *fixed* but the value of the compound is not. Thus, it would be enough to find, say, a case where sentences in the first blank are T, sentences in the second are F but the value of the whole flips from T to F. To show that an operator is not truth functional, any matching combination that makes the whole switch value will do.

To show that an operator is truth functional, we need to show that no such cases are possible. For this, we show *how* the truth value of what is in the blank determines the truth value of the whole. As an example, consider first,

(D)

It is not the case that	____
F	T
T	F

In this table, we represent the truth value of whatever is in the blank by the column under the blank, and the truth value for the whole by the column under the operator. If we put something true according to a consistent story into the blank, the resultant compound is sure to be false according to that story. Thus, for example, in the true story, 'Snow is white', '2 + 2 = 4' and 'Dogs bark' are all true; correspondingly, 'It is not the case that snow is white', 'It is not the case that 2 + 2 = 4' and 'It is not the case that dogs bark' are all false. Similarly, if we put something false according to a story into the blank, the resultant compound is sure to be true according to the story. Thus, for example, in the true story, 'Snow is green' and '2 + 2 = 3' are both false. Correspondingly, 'It is not the case that snow is green' and 'It is not the case that 2 + 2 = 3' are both true. It is no coincidence that the above table for 'It is not the case that ____' looks like the table for $\sim$. We will return to this point shortly.

For a second example of a truth functional operator, consider '____ and ____'. This seems to have table,

(E)

____	and	____
T	***T***	T
T	***F***	F
F	***F***	T
F	***F***	F

Consider a situation where Bob and Sue each love themselves, but hate each other. Then Bob loves Bob and Sue loves Sue is true. But if at least one blank has a sentence that is false, the compound is false. Thus, for example, in that situation, Bob loves Bob and Sue loves Bob is false; Bob loves Sue and Sue loves Sue is false; and Bob

loves Sue and Sue loves Bob is false. For a compound, '____ and ____' to be true, the sentences in both blanks have to be true. And if they are both true, the compound is itself true. So the operator is truth functional. Again, it is no coincidence that the table looks so much like the table for $\wedge$. To show that an operator is truth functional, it is sufficient to produce the table that shows how the truth values of the compound are fixed by the truth values of the sentences in the blanks.

Definitions for Translation

DC A *declarative sentence* is a sentence which has a truth value.

SO A *sentential operator* is an expression containing "blanks" such that when the blanks are filled with declarative sentences, the result is a declarative sentence.

CS Declarative sentences generated from other sentences by means of sentential operators are *compound*; all others are *simple*.

MO The *main operator* of a compound sentence is that operator not in the blank of any other operator.

TF A sentential operator is *truth functional* iff any compound generated by it has its truth value wholly determined by the truth values of the sentences in its blanks.

To show that an operator is not truth functional it is sufficient to produce some situation where truth values for sentences in the blanks are constant, but truth values for the compounds are not.

To show that an operator is truth functional, it is sufficient to produce the table that shows how the truth values of the compound are fixed by truth values of the sentences in the blanks.

For an interesting sort of case, consider the operator 'According to every consistent story ____', and the following attempted table,

	According to every consistent story	____
(F)	?	T
	F	F

(On some accounts, this operator works like 'Necessarily ____'). Say we put some sentence $\mathscr{P}$ that is false according to a consistent story into the blank. Then since $\mathscr{P}$ is false according to that very story, it is not the case that $\mathscr{P}$ according to every consistent story — and the compound is sure to be false. So we fill in the bottom row under the operator as above. So far, so good. But consider 'Dogs bark' and '2 + 2 = 4'. Both are true according to the true story. But only the second is true according to *every* consistent story. So the compound is false with the first in the blank, true with

the second. So 'According to every consistent story ____' is therefore *not* a truth functional operator. The truth value of the compound is not *wholly* determined by the truth value of the sentence in the blank. Similarly, it is natural to think that '____ because ____' is false whenever one of the sentences in its blanks is false. It cannot be true that $\mathscr{P}$ because $\mathscr{Q}$ if not-$\mathscr{P}$, and it cannot be true that $\mathscr{P}$ because $\mathscr{Q}$ if not-$\mathscr{Q}$. If you are not happy, then it cannot be that you are happy because you understand the material; and if you do not understand the material, it cannot be that you are happy because you understand the material. So far, then, the table for '____ because ____' is like the table for '____ and ____'.

	____	because	____
	T	?	T
(G)	T	***F***	F
	F	***F***	T
	F	***F***	F

However, as we saw just above, in contrast to '____ and ____', compounds generated by '____ because ____' may or may not be true when sentences in the blanks are both true. So, although '____ and ____' is truth functional, '____ because ____' is not.

Thus the question is whether we can complete a table of the above sort: If there is a way to complete the table, the operator is truth functional. The test to show an operator is not truth functional simply finds some case to show that such a table cannot be completed.

E5.1. For each of the following, identify the simple sentences that are parts. If the sentence is compound, use underlines to exhibit its operator structure, and say what is its main operator.

a. Bob likes Mary.

b. Jim believes that Bob likes Mary.

c. It is not the case that Bob likes Mary.

d. Jane heard that it is not the case that Bob likes Mary.

e. Jane heard that Jim believes that it is not the case that Bob likes Mary.

f. Voldemort is very powerful, but it is not the case that Voldemort kills Harry at birth.

g. Harry likes his godfather and Harry likes Dumbledore, but it is not the case that Harry likes his uncle.

*h. Hermoine believes that studying is good, and Hermione studies hard, but Ron believes studying is good, and it is not the case that Ron studies hard.

i. Malfoy believes mudbloods are scum, but it is not the case that mudbloods are scum; and Malfoy is a dork.

j. Harry believes that Voldemort is evil and Hermione believes that Voldemort is evil, but it is not the case that Bellatrix believes that Voldemort is evil.

E5.2. Which of the following operators are truth functional and which are not? If the operator is truth functional, display the relevant table; if it is not, give cases that flip the value of the compound, with the value in the blanks constant.

*a. It is a fact that ____

b. Elmore believes that ____

*c. ____ but ____

d. According to some consistent story ____

e. Although ____, ____

*f. It is always the case that ____

g. Sometimes it is the case that ____

h. ____ therefore ____

i. ____ however ____

j. Either ____ or ____ (or both)

5.2.2 Parse Trees

We are now ready to outline a procedure for translation into our formal sentential language. In the end, you will often be able to see how translations should go and to write them down without going through all the official steps. However, the procedure should get you thinking in the right direction, and remain useful for complex cases. To translate some ordinary sentences $\mathcal{P}_1 \ldots \mathcal{P}_n$ the basic translation procedure is,

TP (1) Convert the ordinary $\mathscr{P}_1 \dots \mathscr{P}_n$ into corresponding ordinary equivalents exposing truth functional and operator structure.

(2) Generate a "parse tree" for each of $\mathscr{P}_1 \dots \mathscr{P}_n$ and specify the interpretation function II by assigning sentence letters to sentences at the bottom nodes.

(3) Construct a parallel tree that translates each node from the parse tree, to generate a formal $\mathscr{P}'_i$ for each $\mathscr{P}_i$.

For now at least, the idea behind step (1) is simple: Sometimes all you need to do is expose operator structure by introducing underlines. In complex cases, this can be difficult! But we know how to do this. Sometimes, however, truth functional structure does not lie on the surface. Ordinary sentences are *equivalent* when they are true and false in exactly the same consistent stories. And we want ordinary equivalents exposing truth functional structure. Suppose $\mathscr{P}$ is a sentence of the sort,

(H) Bob is not happy

Is this a truth functional compound? Not officially. There is no declarative sentence in the blank of a sentential operator; so it is not compound; so it is not a truth functional compound. But one might think that (H) is short for,

(I) It is not the case that <u>Bob is happy</u>

which is a truth functional compound. At least, (H) and (I) are equivalent in the sense that they are true and false in the same consistent stories. Similarly, 'Bob and Carol are happy' is not a compound of the sort we have described, because 'Bob' is not a declarative sentence. However, it is a short step from this sentence to the equivalent, '<u>Bob is happy</u> and <u>Carol is happy</u>' which is an official truth functional compound. As we shall see, in some cases, this step can be more complex. But let us leave it at that for now.

Moving to step (2), in a *parse tree* we begin with sentences constructed as in step (1). If a sentence has a *truth functional* main operator, then it branches downward for the sentence(s) in its blanks. If these have truth functional main operators, they branch for the sentences in *their* blanks; and so forth, until sentences are simple or have non-truth functional main operators. Then we construct the interpretation function II by assigning a distinct sentence letter to each distinct sentence at a bottom node from a tree for the original $\mathscr{P}_1 \dots \mathscr{P}_n$.

Some simple examples should make this clear. Say we want to translate a collection of four sentences.

1. Bob is happy

2. Carol is not happy

3. Bob is healthy and Carol is not

4. Bob is happy and John believes that Carol is not healthy

The first is a simple sentence. Thus there is nothing to be done at step (1). And since there is no main operator, the sentence itself is a completed parse tree. The tree is just,

(J) Bob is happy

Insofar as the simple sentence is a complete branch of the tree, it counts as a bottom node of its tree. It is not yet assigned a sentence letter, so we assign it one. B_1: Bob is happy. We select this letter to remind us of the assignment.

The second sentence is not a truth functional compound. Thus in the first stage, 'Carol is not happy' is expanded to the equivalent, 'It is not the case that Carol is happy'. In this case, there is a main operator; since it is truth functional, the tree has some structure.

(K)

It is not the case that Carol is happy

Carol is happy

The bottom node is simple, so the tree ends. 'Carol is happy' is not assigned a letter; so we assign it one. C_1: Carol is happy.

The third sentence is equivalent to, Bob is healthy and it is not the case that Carol is healthy. Again, the operators are truth functional, and the result is a structured tree.

(L)

Bob is healthy and it is not the case that Carol is healthy

Bob is healthy it is not the case that Carol is healthy

Carol is healthy

The main operator is truth functional. So there is a branch for each of the sentences in its blanks. Observe that underlines continue to reflect the structure of *these* sentences (so we "lift" the sentences from their blanks with structure intact). On the left, 'Bob is healthy' has no main operator, so it does not branch. On the right, 'it is not the

case that Carol is healthy' has a truth functional main operator, and so branches. At bottom, we end up with 'Bob is healthy' and 'Carol is healthy'. Neither has a letter, so we assign them ones. B_2: Bob is healthy; C_2: Carol is healthy.

The final sentence is equivalent to, Bob is happy and John believes it is not the case that Carol is healthy. It has a truth functional main operator. So there is a structured tree.

(M)

Bob is happy and John believes it is not the case that Carol is healthy

Bob is happy   John believes it is not the case that Carol is healthy

On the left, 'Bob is happy' is simple. On the right, 'John believes it is not the case that Carol is healthy' is complex. But its main operator is not truth functional. So *it does not branch*. We only branch for sentences in the blanks of truth functional main operators. Given this, we proceed in the usual way. 'Bob is happy' already has a letter. The other does not; so we give it one. J: John believes it is not the case that Carol is healthy.

And that is all. We have now compiled an interpretation function,

II B_1: Bob is happy
C_1: Carol is happy
B_2: Bob is healthy
C_2: Carol is healthy
J: John believes it is not the case that Carol is healthy

Of course, we might have chosen different letters. All that matters is that we have a distinct letter for each distinct sentence. Our intended interpretations are ones that capture available sentential structure, and make the sentence letters true in situations where these sentences are true and false when they are not. In the last case, there is a compulsion to think that we can somehow get down to the simple sentence 'Carol is happy'. But resist temptation! A non-truth functional operator "seals off" that upon which it operates, and forces us to treat the compound as a unit. We do not automatically assign sentence letters to simple sentences, but rather to parts that are *not* truth functional compounds. Simple sentences fit this description. But so do compounds with non-truth-functional main operators.

E5.3. Use our method to expose truth functional structure and produce parse trees for each of the following. Use your trees to produce an interpretation function for the sentences. Hint: pay attention to punctuation as a guide to structure.

a. Bingo is spotted, and Spot can play bingo.

b. Bingo is not spotted, and Spot cannot play bingo.

c. Bingo is spotted, and believes that Spot cannot play bingo.

*d. It is not the case that: Bingo is spotted and Spot can play bingo.

e. It is not the case that: Bingo is not spotted and Spot cannot play bingo.

E5.4. Use our method to expose truth functional structure and produce parse trees for each of the following. Use your trees to produce an interpretation function for the sentences.

*a. People have rights and dogs have rights, but rocks do not.

b. It is not the case that: rocks have rights, but people do not.

c. Aliens believe that rocks have rights, but it is not the case that people believe it.

d. Aliens landed in Roswell NM in 1947, and live underground but not in my backyard.

e. Rocks do not have rights and aliens do not have rights, but people and dogs do.

5.2.3 Formal Sentences

Now we are ready for step (3) of the translation procedure TP. Our aim is to generate translations by constructing a parallel tree where the force of ordinary truth functional operators is captured by *equivalent* formal operators. An ordinary truth functional operator has a table. Similarly, our formal expressions have tables. Say an ordinary truth functional operator is *equivalent* to some formal expression containing blanks just in case their tables are the same. Thus '$\sim$____' is equivalent to 'it is not the case that ____'. They are equivalent insofar as in each case, the whole has the opposite truth value of what is in the blank. Similarly, '____ $\wedge$ ____' is equivalent to '____ and ____'. In either case, when sentences in the blanks are both T the whole is T, and in other cases, the whole is F. Of course, the complex '$\sim$(____ $\rightarrow$ $\sim$____)' takes the same values as the '____ $\wedge$ ____' that abbreviates it. So different formal expressions may be equivalent to a given ordinary one.

To see how this works, let us return to the sample sentences from above. Again, the idea is to generate a parallel tree. We begin by *using* the sentence letters from our interpretation function for the bottom nodes. The case is particularly simple when the tree has no structure. 'Bob is happy' had a simple unstructured tree, and we assigned it a sentence letter directly. Thus our original and parallel trees are,

(N) Bob is happy $\qquad B_1$

So for a simple sentence, we simply read off the final translation from the interpretation function. So much for the first sentence.

As we have seen, the second sentence is equivalent to 'It is not the case that Carol is happy' with a parse tree as on the left below. We begin the parallel tree on the other side.

(O)

It is not the case that Carol is happy
|
Carol is happy $\qquad C_1$

We know how to translate the bottom node. But now we want to capture the force of the truth functional operator with some equivalent formal operator(s). For this, we need a formal expression containing blanks whose table mirrors the table for the sentential operator in question. In this case, '$\sim$____' works fine. That is, we have,

$\sim$	____
F	T
T	F

It is not the case that	____
F	T
T	F

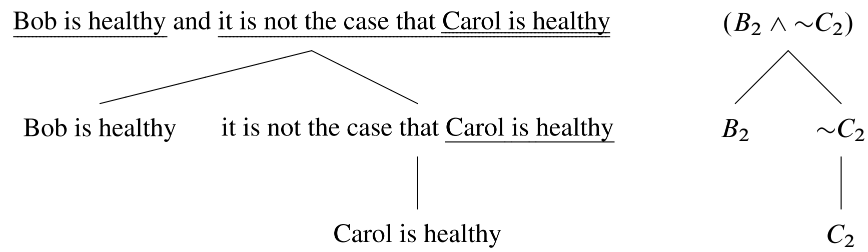
In each case, when the expression in the blank is T, the whole is F, and when the expression in the blank is F, the whole is T. So '$\sim$____' is sufficient as a translation of 'It is not the case that ____'. Other formal expressions might do just as well. Thus, for example, we might go with, '$\sim\sim\sim$____'. The table for this is the same as the table for '$\sim$____'. But it is hard to see why we would do this, with $\sim$ so close at hand. Now the idea is to apply the equivalent operator *to* the already translated expression from the blank. But this is easy to do. Thus we complete the parallel tree as follows.

It is not the case that Carol is happy $\qquad \sim C_1$
| $\qquad$ |
Carol is happy $\qquad C_1$

The result is the completed translation, $\sim C_1$.

The third sentence has a parse tree as on the left, and resultant parallel tree as on the right. As usual, we begin with sentence letters from the interpretation function for the bottom nodes.

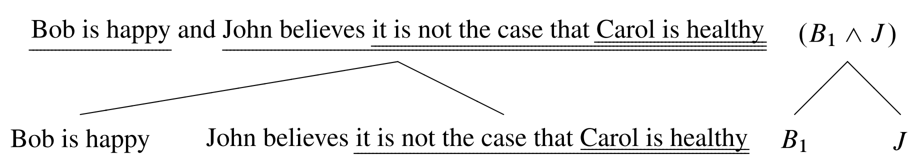
(P)



Given translations for the bottom nodes, we work our way through the tree, applying equivalent operators to translations already obtained. As we have seen, a natural translation of 'it is not the case that ____' is '$\sim$____'. Thus, working up from 'Carol is healthy', our parallel to 'it is not the case that Carol is healthy' is $\sim C_2$. But now we have translations for both of the blanks of '____ and ____'. As we have seen, this has the same table as '(____ $\wedge$ ____)'. So that is our translation. Again, other expressions might do. In particular, $\wedge$ is an abbreviation with the same table as '$\sim$(____ $\rightarrow$ $\sim$____)'. In each case, the whole is true when the sentences in both blanks are true, and otherwise false. Since this is the same as for '____ and ____', either would do as a translation. But again, the simplest thing is to go with '(____ $\wedge$ ____)'. Thus the final result is $(B_2 \wedge \sim C_2)$. With the alternate translation for the main operator, the result would have been $\sim(B_2 \rightarrow \sim\sim C_2)$. Observe that the parallel tree is an upside-down version of the (by now quite familiar) tree by which we would show that the expression is a sentence.

Our last sentence is equivalent to, Bob is happy and John believes it is not the case that Carol is healthy. Given what we have done, the parallel tree should be easy to construct.

(Q)



Given that the tree "bottoms out" on both 'Bob is happy' and 'John believes it is not the case that Carol is healthy' the only operator to translate is the main operator '____ and ____'. And we have just seen how to deal with that. The result is the completed translation, $(B_1 \wedge J)$.

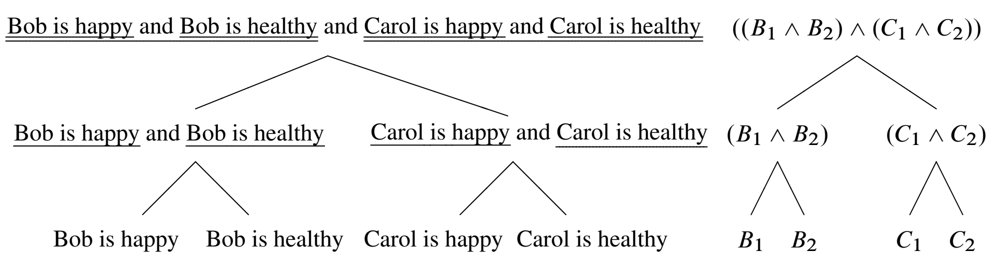
Again, once you become familiar with this procedure, the full method, with the trees, may become tedious — and we will often want to set it to the side. But notice:

the method breeds good habits! And the method puts us in a position to translate complex expressions, even ones that are so complex that we can barely grasp what they are saying. Beginning with the main operator, we break expressions down from complex parts to ones that are simpler. Then we construct translations, one operator at a time, where each step is manageable. Also, we should be able to see *why* the method results in good translations: For any situation and corresponding intended interpretation, truth values for *basic* parts are the same by the specification of the interpretation function. And given that operators are equivalent, truth values for parts built out of them must be the same as well, all the way up to the truth value of the whole. We satisfy the first part of our criterion CG insofar as the way we break down sentences in parse trees forces us to capture all the truth functional structure there is to be captured.

For a last example, consider, 'Bob is happy and Bob is healthy and Carol is happy and Carol is healthy'. This is true only if 'Bob is happy', 'Bob is healthy', 'Carol is happy', and 'Carol is healthy' are all true. But the method may apply in different ways. We might at step one, treat the sentence as a complex expression involving multiple uses of '____ and ____'; perhaps something like,

(R) Bob is happy and Bob is healthy and Carol is happy and Carol is healthy

In this case, there is a straightforward move from the ordinary operators to formal ones in the final step. That is, the situation is as follows.



So we use multiple applications of our standard caret operator. But we might have treated the sentence as something like,

(S) Bob is happy and Bob is healthy and Carol is happy and Carol is healthy

involving a single four-blank operator, '____ and ____ and ____ and ____', which yields true only when sentences in all its blanks are true. We have not seen anything like this before, but nothing stops a tree with four branches all at once. In this case, we would begin,

Bob is happy and Bob is healthy and Carol is happy and Carol is healthy

Bob is happy Bob is healthy Carol is happy Carol is healthy B_1 B_2 C_1 C_2

But now, for an equivalent operator we need a formal expression with *four* blanks that is true when sentences in all the blanks are true and otherwise false. Here is something that would do: '((____ $\wedge$ ____) $\wedge$ (____ $\wedge$ ____))'. On either of these approaches, then, the result is $((B_1 \wedge B_2) \wedge (C_1 \wedge C_2))$. Other options might result in something like $(((B_1 \wedge B_2) \wedge C_1) \wedge C_2)$. In this way, there is room for shifting burden between steps one and three. Such shifting explains how step (1) can be more complex than it was initially represented to be. Choices about expanding truth functional structure in the initial stage, may matter for what are the equivalent operators at the end. And the case exhibits how there are options for different, equally good, translations of the same ordinary expressions. What matters for CG is that resultant expressions capture available structure and be true when the originals are true and false when the originals are false. In most cases, one translation will be more *natural* than others, and it is good form to strive for natural translations. If there had been a comma so that the original sentence was, 'Bob is happy and Bob is healthy, and Carol is happy and Carol is healthy' it would have been most natural to go for an account along the lines of (R). And it is crazy to use, say, '$\sim\sim\sim$____' when '$\sim$____' will do as well.

*E5.5. Construct parallel trees to complete the translation of the sentences from E5.3 and E5.4. Hint: you will not need any operators other than $\sim$ and $\wedge$.

E5.6. Use our method to translate each of the following. That is, generate parse trees with an interpretation function for all the sentences, and then parallel trees to produce formal equivalents.

a. Plato and Aristotle were great philosophers, but Ayn Rand was not.

b. Plato was a great philosopher, and everything Plato said was true, but Ayn Rand was not a great philosopher, and not everything she said was true.

*c. It is not the case that: everything Plato, and Aristotle, and Ayn Rand said was true.

d. Plato was a great philosopher but not everything he said was true, and Aristotle was a great philosopher but not everything he said was true.

e. Not everyone agrees that Ayn Rand was not a great philosopher, and not everyone thinks that not everything she said was true.

E5.7. Use our method to translate each of the following. That is, generate parse trees with an interpretation function for all the sentences, and then parallel trees to produce formal equivalents.

a. Bob and Sue and Jim will pass the class.

b. Sue will pass the class, but it is not the case that: Bob will pass and Jim will pass.

c. It is not the case that: Bob will pass the class and Sue will not.

d. Jim will not pass the class, but it is not the case that: Bob will not pass and Sue will not pass.

e. It is not the case that: Jim will pass and not pass, and it is not the case that: Sue will pass and not pass.

5.2.4 And, Or, Not

Our idea has been to recognize when truth conditions for ordinary and formal sentences are the same. As we have seen, this turns out to require recognizing when *operators* have the same tables. We have had a lot to say about 'it is not the case that ____' and '____ and ____'. We now turn to a more general treatment. We will not be able to provide a complete menu of ordinary operators. Rather, we will see that some uses of some ordinary operators can be appropriately translated by our symbols. We should be able to discuss enough cases for you to see how to approach others on a case-by-case basis. The discussion is organized around our operators, $\sim$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$, taken in that order.

First, as we have seen, 'It is not the case that ____' has the same table as $\sim$. And various ordinary expressions may be equivalent to expressions involving this operator. Thus, 'Bob is not married' and 'Bob is unmarried' might be understood as equivalent to 'It is not the case that <u>Bob is married</u>'. Given this, we might assign a sentence letter, say, M to 'Bob is married' and translate $\sim M$. But the second case calls for comment. By comparison, consider, 'Bob is unlucky'. Given what we have done, it is natural to treat 'Bob is unlucky' as equivalent to 'It is not the case that <u>Bob is lucky</u>'; assign L to 'Bob is lucky'; and translate $\sim L$. But this is not obviously right. Consider three situations: (i) Bob goes to Las Vegas with $1,000, and comes

away with $1,000,000. (ii) Bob goes to Las Vegas with $1,000, and comes away with $100, having seen a show and had a good time. (iii) Bob goes to Las Vegas with $1,000, falls into a manhole on his way into the casino, and has his money stolen by a light-fingered thief on the way down. In the first case he is lucky; in the third, unlucky. But, in the second, one might want to say that he was neither lucky nor unlucky. If this is right, 'Bob is unlucky' is *not* equivalent to 'It is not the case that Bob is lucky' — for it is not the case that Bob is lucky in *both* situations (ii) and (iii). Thus we might have to assign 'Bob is lucky' one letter, and 'Bob is unlucky' another.[1] Decisions about this sort of thing may depend heavily on context, and assumptions which are in the background of conversation. We will ordinarily *assume* contexts where there is no "neutral" state — so that being unlucky is not being lucky, and similarly in other cases.

Second, as we have seen, '____ and ____' has the same table as $\wedge$. As you may recall from E5.2, another common operator that works this way is '____ but ____'. Consider, for example, 'Bob likes Mary but Mary likes Jim'. Suppose Bob does like Mary and Mary likes Jim; then the compound sentence is true. Suppose one of the simples is false, Bob does not like Mary or Mary does not like Jim; then the compound is false. Thus '____ but ____' has the table,

(T)

____	but	____
T	***T***	T
T	***F***	F
F	***F***	T
F	***F***	F

and so has the same table as $\wedge$. So, in this case, we might assign B to 'Bob likes Mary' M to 'Mary likes Jim', and translate, $(B \wedge M)$. Of course, the ordinary expression 'but' carries a sense of opposition that 'and' does not. Our point is not that 'and' and 'but' somehow *mean* the same, but rather that compounds formed by means of them are true and false under the same truth functional conditions. Another common operator with this table is 'Although ____, ____'. You should convince yourself that this is so, and be able to find other ordinary terms that work just the same way.

Once again, however, there is room for caution in some cases. Consider, for example, 'Bob took a shower and got dressed'. Given what we have done, it is

[1] Or so we have to do in the context of our logic where T and F are the only truth values. Another option is to allow three values so that the one letter might be T, F or neither. It is possible to proceed on this basis — though the two valued (classical) approach has the virtue of relative simplicity! With the classical approach as background, some such alternatives are developed in Priest, *Non-Classical Logics*.

natural to treat this as equivalent to 'Bob took a shower and Bob got dressed'; assign letters S and D; and translate $(S \wedge D)$. But this is not obviously right. Suppose Bob gets dressed, but then realizes that he is late for a date and forgot to shower, so he jumps in the shower fully clothed, and air-dries on the way. Then it is true that Bob took a shower, and true that Bob got dressed. But is it true that Bob took a shower and got dressed? If not — because the order is wrong — our translation $(S \wedge D)$ might be true when the original sentence is not. Again, decisions about this sort of thing depend heavily upon context and background assumptions. And there may be a distinction between what is *said* and what is conversationally *implied* in a given context. Perhaps what was said corresponds to the table, so that our translation is right, though there are certain assumptions typically made in conversation that go beyond. But we need not get into this. Our point is not that the ordinary 'and' *always* works like our operator $\wedge$; rather the point is that some (indeed, many) ordinary uses are rightly regarded as having the same table.[2] Again, we will ordinarily *assume* a context where 'and', 'but' and the like have tables that correspond to $\wedge$.

Now consider 'Neither Bob likes Sue nor Sue likes Bob'. This seems to involve an operator, 'Neither ____ nor ____' with the following table.

(U)

Neither	____ nor	____
F	T	T
F	T	F
F	F	T
T	F	F

'Neither Bob likes Sue nor Sue likes Bob' is true just when 'Bob likes Sue' and 'Sue likes Bob' are both false, and otherwise false. But no operator of our formal language has a table which is T just when components are both F. Still, we may form complex expressions which work this way. Thus, for example, '$(\sim$____ $\wedge \sim$____$)$' is T just when sentences in the blanks are both F.

[2]The ability to make this point is an important byproduct of our having introduced the formal operators "as themselves." Where $\wedge$ and the like are introduced as *being* direct translations of ordinary operators, a natural reaction to cases of this sort — a reaction had even by some professional logicians and philosophers — is that "the table is wrong." But this is mistaken! $\wedge$ has its own significance, which may or may not agree with the shifting meaning of ordinary terms. The situation is no different than for translation across ordinary languages, where terms may or may not have uniform equivalents.

But now, one may feel a certain tension with our account of what it is for an operator to be truth functional — for there seem to be contexts where the truth value of sentences in the blanks does not determine the truth value of the whole, even for a purportedly truth functional operator like '____ and ____'. However, we want to distinguish different *senses* in which an operator may be used (or an *ambiguity*, as between a *bank* of a river, and a *bank* where you deposit money), so that when an operator is used with just one sense it has some definite truth function.

(V)

$\mathscr{P}$	$\mathscr{Q}$	$\sim$	$\mathscr{P}$	$\wedge$	$\sim$	$\mathscr{Q}$
T	T	F	T	***F***	F	T
T	F	F	T	***F***	T	F
F	T	T	F	***F***	F	T
F	F	T	F	***T***	T	F

So ‘($\sim$____ $\wedge$ $\sim$____)’ is a good translation of ‘Neither ____ nor ____’. Another expression with the same table is $\sim(\mathscr{P} \vee \mathscr{Q})$. As it turns out, for any table a truth functional operator may have, there is some way to generate that table by means of our formal operators — and in fact, by means of just the operators $\sim$ and $\wedge$, or just the operators $\sim$ and $\vee$, or just the operators $\sim$ and $\rightarrow$. We will prove this in Part III. For now, let us return to our survey of expressions which do correspond to operators.

The operator which is most naturally associated with $\vee$ is ‘____ or ____’. In this case, there is room for caution from the start. Consider first a restaurant menu which says that you will get soup, or you will get salad, with your dinner. This is naturally understood as ‘you will get soup or you will get salad’ where the sentential operator is ‘____ or ____’. In this case, the table would seem to be,

(W)

____	or	____
T	***F***	T
T	***T***	F
F	***T***	T
F	***F***	F

The compound is true if you get soup, true if you get salad, but not if you get neither or both. None of our operators has this table.

But contrast this case with one where a professor promises either to give you an ‘A’ on a paper, or to give you very good comments so that you will know what went wrong. Suppose the professor gets excited about your paper, giving you both an ‘A’ and comments. Presumably, she did not break her promise! That is, in this case, we seem to have, ‘I will give you an ‘A’ or I will give you comments’ with the table,

(X)

____	or	____
T	***T***	T
T	***T***	F
F	***T***	T
F	***F***	F

The professor breaks her word just in case she gives you a low grade without comments. This table is identical to the table for $\vee$. For another case, suppose you set out to buy a power saw, and say to your friend ‘I will go to Home Depot or I will go Lowes’. You go to Home Depot, do not find what you want, so go to Lowes and make your purchase. When your friend later asks where you went, and you say you

went to both, he or she will not say you lied (!) when you said where you were going — for your statement required only that you would try at least one of those places.

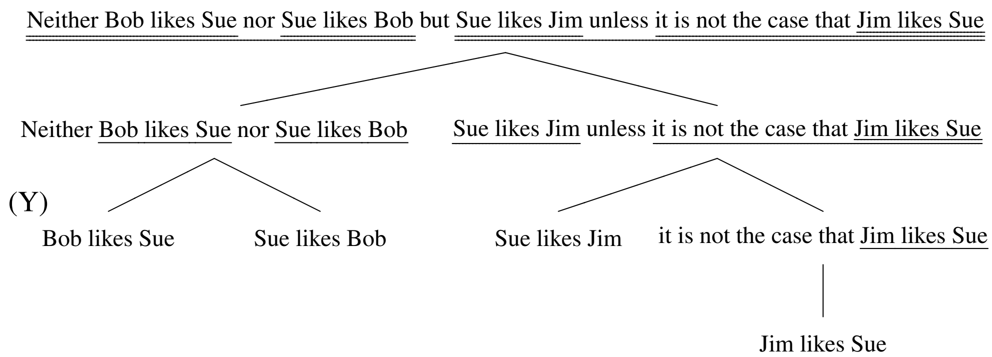
The grading and shopping cases represent the so-called "inclusive" use of 'or' — including the case when both components are T; the menu uses the "exclusive" use of 'or' — excluding the case when both are T. Ordinarily, we will *assume* that 'or' is used in its inclusive sense, and so is translated directly by $\vee$.[3] Another operator that works this way is '____ unless ____'. Again, there are exclusive and inclusive senses — which you should be able to see by considering restaurant and grade examples as above. And again, we will ordinarily assume that the inclusive sense is intended. For the exclusive cases, we can generate the table by means of complex expressions. Thus, for example both $(\mathcal{P} \leftrightarrow \sim\mathcal{Q})$ and $[(\mathcal{P} \vee \mathcal{Q}) \wedge \sim(\mathcal{P} \wedge \mathcal{Q})]$ do the job. You should convince yourself that this is so.

Observe that 'either ____ or ____' says the same as '____ or ____'; and 'both ____ and ____' the same as '____ and ____'. So one might think that 'either' and 'both' have no real role. They do however serve a sort of "bracketing" function. So for example one way to think about 'neither ____ nor ____' is as a *negation* of 'either ____ or ____' (the 'n' to indicate negation). Then observe that 'neither Bob nor Sue is happy' is not legitimately parsed into 'it is not the case that either Bob is happy or Sue is happy' with main operator '____ or ____', insofar 'either Bob is happy' in the blank of 'it is not the case that ____'is not a complete sentence. The required result is 'it is not the case that either Bob is happy or Sue is happy' with complete sentences in each blank and translation $\sim(B \vee S)$ — where this has the same table as $\sim B \wedge \sim S$, the translation suggested above. A similar bracketing results from 'both ____ and ____'. Thus the proper understanding of 'not both Bob and Sue are happy' is 'it is not the case that both Bob is happy and Sue is happy' with translation, $\sim(B \wedge S)$. So 'either' and 'both' bracket what comes after.

And we continue to work with complex forms on trees. Thus, for example, consider 'Neither Bob likes Sue nor Sue likes Bob, but Sue likes Jim unless Jim does not like her'. This is a mouthful, but we can deal with it in the usual way. The hard part, perhaps, is just exposing the operator structure.

---

[3] Again, there may be a distinction between what is *said* and what is conversationally *implied* in a given context. Perhaps what was said generally corresponds to the inclusive table, though many uses are against background assumptions which automatically exclude the case when both are T. But we need not get into this. It is enough that some uses are according to the inclusive table.



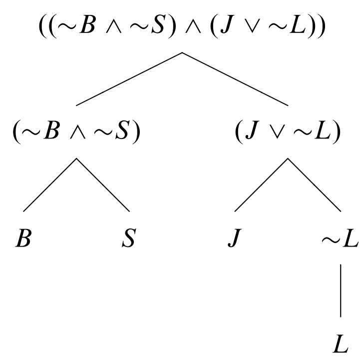
Given this, with what we have said above, generate the interpretation function and then the parallel tree as follows.

B: Bob likes Sue

S: Sue likes Bob

J: Sue likes Jim

L: Jim likes Sue



We have seen that ‘(____ $\vee$ ____)’ is equivalent to ‘____ unless ____’; that ‘($\sim$____ $\wedge$ $\sim$____)’ is equivalent to ‘neither ____ nor ____’; and that ‘(____ $\wedge$ ____)’ is equivalent to ‘____ but ____’. Given these, everything works as before. Again, the complex problem is rendered simple, if we attack it one operator at a time. Another natural option would be $(\sim(B \vee S) \wedge (J \vee \sim L))$ with the alternate version of ‘neither ____ nor ____’.

E5.8. Using the interpretation function below, produce parse trees and then parallel ones to complete the translation for each of the following.

B: Bob likes Sue

S: Sue likes Bob

B_1: Bob is cool

S_1: Sue is cool

a. Bob likes Sue.

b. Sue does not like Bob.

c. Bob likes Sue and Sue likes Bob.

d. Bob likes Sue or Sue likes Bob.

e. Bob likes Sue unless she is not cool.

f. Either Bob does not like Sue or Sue does not like Bob.

g. Neither Bob likes Sue, nor Sue likes Bob.

*h. Not both Bob and Sue are cool.

i. Bob and Sue are cool, and Bob likes Sue, but Sue does not like Bob.

j. Although neither Bob nor Sue are cool, either Bob likes Sue, or Sue likes Bob.

E5.9. Use our method to translate each of the following. That is, generate parse trees with an interpretation function for all the sentences, and then parallel trees to produce formal equivalents.[4]

a. Harry is not a Muggle.

b. Neither Harry nor Hermione are Muggles.

c. Either Harry's or Hermione's parents are Muggles.

*d. Neither Harry, nor Ron, nor Hermione are Muggles.

e. Not both Harry and Hermione have Muggle parents.

f. The game of Quidditch continues unless the Snitch is caught.

*g. Although blatching and blagging are illegal in Quidditch, the woolongong shimmy is not.

h. Either the beater hits the bludger or you are not protected from it, and the bludger is a very heavy ball.

[4]My source for the information on Quidditch is Kennilworthy Whisp (aka, J.K. Rowling), *Quidditch Through the Ages*, along with a daughter who is a rabid fan of all things Potter.

i. The Chudley Cannons are not the best Quidditch team ever, however they hope for the best.

j. Harry won the Quidditch cup in his 3rd year at Hogwarts, but not in his 1st, 2nd, 4th, or 5th.

5.2.5 If, Iff

The operator which is most naturally associated with $\rightarrow$ is ‘if ____ then ____’. Consider some fellow, perhaps of less than sterling character, of whom we assert, ‘If he loves her, then she is rich’. In this case, the table begins,

(Z)

If ____	then	____
T	***T***	T
T	***F***	F
F	?	T
F	***T***	F

If ‘He loves her’ and ‘She is rich’ are both true, then what we said about him is true. If he loves her, but she is not rich, what we said was wrong. If he does not love her, and she is poor, then we are also fine, for all we said was that *if* he loves her, then she is rich. But what about the other case? Suppose he does not love her, but she is rich. There is a temptation to say that our conditional assertion is false. But do not give in! Notice: we did not say that he loves all the rich girls. All we said was that *if* he loves this particular girl, then she is rich. So the existence of rich girls he does not love does not undercut our claim. For another case, say you are trying to find the car he is driving and say ‘If he is in his own car, then it is a Corvette.’ That is, ‘If he is in his own car then it is a Corvette’. You would be mistaken if he has traded his Corvette for a Yugo. But say the Corvette is in the shop and he is driving a loaner that also happens to be a Corvette. Then ‘He is in his own car’ is F and ‘He is driving a Corvette’ is T. Still, there is nothing wrong with your claim — *if* he is in his own car, then it is a Corvette. Given this, we are left with the completed table,

(AA)

If ____	then	____
T	***T***	T
T	***F***	F
F	***T***	T
F	***T***	F

which is identical to the table for $\rightarrow$. With L for ‘He loves her’ and R for ‘She is rich’, for ‘If he loves her then she is rich’ the natural translation is $(L \rightarrow R)$. Another case which works this way is He loves her only if she is rich. You should think through this as above. So far, perhaps, so good.

But the conditional calls for special comment. First, notice that the table shifts with the position of 'if'. Suppose he loves her if she is rich. Intuitively, this says the same as, 'If she is rich then he loves her'. This time, we are mistaken if she is rich and he does not love her. Thus, with the above table and assignments, we end up with translation $(R \rightarrow L)$. Notice that the order is switched around the arrow. We can make this point directly from the original claim.

(AB)

he loves her	if	she is rich
T	***T***	T
T	***T***	F
F	***F***	T
F	***T***	F

The claim is false just in the case where she is rich but he does not love her. The result is *not* the same as the table for $\rightarrow$. What we need is an expression that is F in the case when L is F and R is T, and otherwise T. We get just this with $(R \rightarrow L)$. Of course, this is just the same result as by intuitively reversing the operator into the regular 'If ____ then ____' form.

In the formal language, the *order* of the components is crucial. In a true material conditional, the truth of the antecedent guarantees the truth of the consequent. In ordinary language, this role is played, not by the order of the components, but by operator placement. In general, *if* by itself is an *antecedent* indicator; and *only if* is a *consequent* indicator. That is, we get,

(AC)

If $\mathcal{P}$ then $\mathcal{Q}$	$\Longrightarrow$	$(\mathcal{P} \rightarrow \mathcal{Q})$
$\mathcal{P}$ if $\mathcal{Q}$	$\Longrightarrow$	$(\mathcal{Q} \rightarrow \mathcal{P})$
$\mathcal{P}$ only if $\mathcal{Q}$	$\Longrightarrow$	$(\mathcal{P} \rightarrow \mathcal{Q})$
only if $\mathcal{P}$, $\mathcal{Q}$	$\Longrightarrow$	$(\mathcal{Q} \rightarrow \mathcal{P})$

'If', taken alone, identifies what does the guaranteeing, and so the antecedent of our material conditional; 'only if' identifies what is guaranteed, and so the consequent.[5]

As we have just seen, the natural translation of '$\mathcal{P}$ if $\mathcal{Q}$' is $\mathcal{Q} \rightarrow \mathcal{P}$, and the translation of '$\mathcal{P}$ only if $\mathcal{Q}$' is $\mathcal{P} \rightarrow \mathcal{Q}$. Thus it should come as no surprise that the translation of '$\mathcal{P}$ if *and* only if $\mathcal{Q}$' is $(\mathcal{P} \rightarrow \mathcal{Q}) \wedge (\mathcal{Q} \rightarrow \mathcal{P})$, where this is precisely what is abbreviated by $(\mathcal{P} \leftrightarrow \mathcal{Q})$. We can also make this point directly. Consider, 'he loves her if and only if she is rich'. The operator is truth functional, with the table,

[5]It may feel natural to convert '$\mathcal{P}$ unless $\mathcal{Q}$' to '$\mathcal{P}$ if not $\mathcal{Q}$' and translate $(\sim\mathcal{Q} \rightarrow \mathcal{P})$. This is fine and, as is clear from the abbreviated form, equivalent to $(\mathcal{Q} \vee \mathcal{P})$. However, with the extra negation and concern about direction of the arrow, it is easy to get confused on this approach — so the simple wedge is less likely to go wrong.

Cause and Conditional

It is important that the material conditional does *not* directly indicate causal connection. Suppose we have sentences S: You strike the match, and L: The match will light. And consider,

(i)	If you strike the match then it will light	$S \to L$
(ii)	The match will light only if you strike it	$L \to S$

with natural translations by our method on the right. Good. But, clearly the *cause* of the lighting is the striking. So the first arrow runs from cause to effect, and the second from effect to cause. Why? In (i) we represent the cause as *sufficient* for the effect: striking the match guarantees that it will light. In (ii) we represent the cause as *necessary* for the effect — the only way to get the match to light, is to strike it — so that the match's lighting guarantees that it was struck.

There may be a certain *tendency* to associate the ordinary 'if' and 'only if' with cause, so that we say, 'if $\mathcal{P}$ then $\mathcal{Q}$' when we think of $\mathcal{P}$ as a (sufficient) cause of $\mathcal{Q}$, and say '$\mathcal{P}$ only if $\mathcal{Q}$' when we think of $\mathcal{Q}$ as a (necessary) cause of $\mathcal{P}$. But causal direction is not reflected by the arrow, which comes out $(\mathcal{P} \to \mathcal{Q})$ either way. The material conditional indicates *guarantee*.

This point is important insofar as certain ordinary conditionals seem inextricably tied to causation. This is particularly the case with "subjunctive" conditionals (conditionals about what *would* have been). Suppose I was playing basketball and said, 'If I had played Kobe, I would have won' where this is, 'If it were the case that I played Kobe then it would have been the case that I won the game'. Intuitively, this is false, Kobe would wipe the floor with me. But contrast, 'If it were the case that I played Lassie then it would have been the case that I won the game'. Now, intuitively, this is true; Lassie has many talents but, presumably, basketball is not among them — and I could take her. But I have never played Kobe or Lassie, so both 'I played Kobe' and 'I played Lassie' are false. Thus the truth value of the whole conditional changes from false to true though the values of sentences in the blanks remain the same; and 'If it were the case that ____ then it would have been the case that ____' is not even truth functional. Subjunctive conditionals do offer a sort of guarantee, but the guarantee is for situations alternate to the way things actually are. So actual truth values do not determine the truth of the conditional.

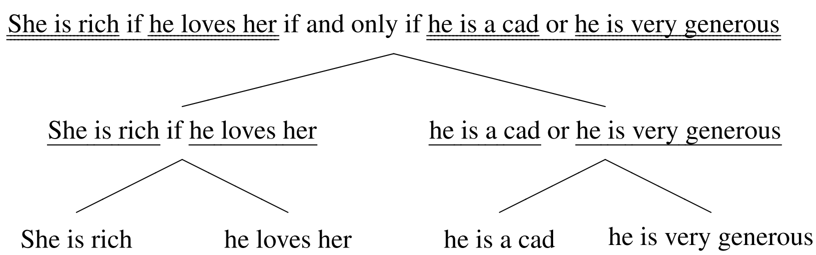
Conditionals other than the material conditional are a central theme of Priest, *Non-Classical Logics*. As usual, we simply assume that 'if' and 'only if' are used in their truth functional sense, and so are given a good translation by $\to$.

	he loves her	if and only if	she is rich
	T	***T***	T
(AD)	T	***F***	F
	F	***F***	T
	F	***T***	F

It cannot be that he loves her and she is not rich, because he loves her *only if* she is rich; so the second row is F. And it cannot be that she is rich and he does not love her, because he loves her *if* she is rich; so the third row is F. The conditional is true just when both she is rich and he loves her, or neither. Another operator that works this way is '____ just in case ____'. You should convince yourself that this is so. Notice that 'if', 'only if', and 'if and only if' play very different roles for translation — you almost want to think of them as completely different words: *if*, *onlyif*, and *ifandonlyif*, each with their own distinctive logical role. Do not get the different roles confused!

For an example that puts some of this together, consider, 'She is rich if he loves her, if and only if he is a cad or very generous'. This comes to the following.

(AE)



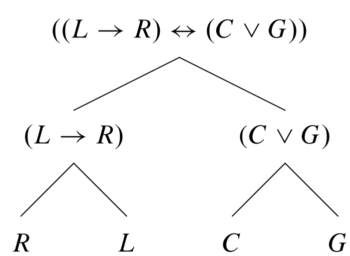
We begin by assigning sentence letters to the simple sentences at the bottom. Then the parallel tree is constructed as follows.

R: She is rich

L: He loves her

C: He is a cad

G: He is very generous



Observe that she is rich if he loves her is equivalent to $(L \rightarrow R)$, not the other way around. Then the wedge translates '____ or ____', and the main operator has the same table as $\leftrightarrow$.

Notice again that our procedure for translating, one operator or part at a time, lets us translate even where the original is so complex that it is difficult to compre-

hend. The method forces us to capture all available truth functional structure, and the translation is thus good insofar as given the specified interpretation function, the method makes the formal sentence true at just the consistent stories where the original is true. It does this because the formal and informal sentences *work* the same way. Eventually, you want to be able to work translations without the trees! (And maybe you have already begun to do so.) In fact, it will be helpful to generate them from the *top down*, rather than from the bottom up, building the translation operator-by-operator as you take the sentence apart from the main operator. But, of course, the result should be the same no matter how you do it.

From definition AR on p. 4 an argument is some sentences, one of which (the conclusion) is taken to be supported by the remaining sentences (the premises). In some courses on logic or critical reasoning, one might spend a great deal of time learning to identify premises and conclusions in ordinary discourse. However, we have taken this much as given, representing arguments in *standard form*, with premises listed as complete sentences above a line, and the conclusion under. Thus, for example,

(AF)
If you strike the match, then it will light
The match will not light
———
You did not strike the match

is a simple argument of the sort we might have encountered in chapter 1. To translate the argument, we produce a translation for the premises and conclusion, retaining the "standard-form" structure. Thus as in the discussion of causation on p. 164, we might end up with an interpretation function and translation as below,

S: You strike the match	$S \rightarrow L$
L: The match will light	$\sim L$
	———
	$\sim S$

The result is an object to which we can apply our semantic and derivation methods in a straightforward way.

And this is what we have been after: If a formal argument is sententially valid, then the corresponding ordinary argument must be logically valid. For some good formal translation of its premises and conclusion, suppose an argument is sententially valid; then by SV there is *no* interpretation on which the premises are true and the conclusion is false; so there is no *intended* interpretation on which the premises are true and the conclusion is false; but given a good translation, by CG, the ordinary-language premises and conclusion have the same truth values at any consistent story as formal expressions on the corresponding intended interpretation; so no *consistent*

story has the premises true and the conclusion false; so by LV the original argument is logically valid. We will make this point again, in some detail, in Part III. For now, notice that our formal methods, derivations and truth tables, apply to arguments of arbitrary complexity. So we are in a position to demonstrate validity for arguments that would have set us on our heels in chapter 1. With this in mind, consider again the butler case (B) that we began with from p. 2. The demonstration that the argument is logically valid is entirely straightforward, by a good translation and then truth tables to demonstrate semantic validity. (It remains for Part III to show how *derivations* matter for semantic validity.)

E5.10. Using the interpretation function below, produce parse trees and then parallel ones to complete the translation for each of the following.

L: Lassie barks

T: Timmy is in trouble

P: Pa will help

H: Lassie is healthy

a. If Timmy is in trouble, then Lassie barks.

b. Timmy is in trouble if Lassie barks.

c. Lassie barks only if Timmy is in trouble.

d. If Timmy is in trouble and Lassie barks, then Pa will help.

*e. If Timmy is in trouble, then if Lassie barks Pa will help.

f. If Pa will help only if Lassie barks, then Pa will help if and only if Timmy is in trouble.

g. Pa will help if Lassie barks, just in case Lassie barks only if Timmy is in trouble.

h. If Timmy is in trouble and Pa does not help, then Lassie is not healthy or does not bark.

*i. If Timmy is in trouble, then either Lassie is not healthy or if Lassie barks then Pa will help.

j. If Lassie neither barks nor is healthy, then Timmy is in trouble if Pa will not help.

E5.11. Use our method, with or without parse trees, to produce a translation, including interpretation function for the following.

a. If animals feel pain, then animals have intrinsic value.

b. Animals have intrinsic value only if they feel pain.

c. Although animals feel pain, vegetarianism is not right.

d. Animals do not have intrinsic value unless vegetarianism is not right.

e. Vegetarianism is not right only if animals do not feel pain or do not have intrinsic value.

f. If you think animals feel pain, then vegetarianism is right.

*g. If you think animals do not feel pain, then vegetarianism is not right.

h. If animals feel pain, then if animals have intrinsic value if they feel pain, then animals have intrinsic value.

*i. Vegetarianism is right only if both animals feel pain, and animals have intrinsic value just in case they feel pain; but it is not the case that animals have intrinsic value just in case they feel pain.

j. If animals do not feel pain if and only if you think animals do not feel pain, but you do think animals feel pain, then you do not think that animals feel pain.

E5.12. For each of the following arguments: (i) Produce a good translation, including interpretation function and translations for the premises and conclusion. Then (ii) use truth tables to determine whether the argument is sententially valid.

*a. Our car will not run unless it has gasoline
Our car has gasoline
———
Our car will run

b. If Bill is president, then Hillary is first lady
Hillary is not first lady
———
Bill is not president

c. Snow is white and snow is not white
———
Dogs can fly

d. If Mustard murdered Boddy, then it happened in the library.
The weapon was the pipe if and only if it did not happen in the library, and the weapon was not the pipe only if Mustard murdered him
———
Mustard murdered Boddy

e. There is evil
If god is good, there is no evil unless he has an excuse for allowing it.
If god is omnipotent, then he does not have an excuse for allowing evil.
———
God is not both good and omnipotent.

E5.13. For each of the arguments in E512 that is sententially valid, produce a derivation to show that it is valid in *AD*.

E5.14. Use translation and truth tables to show that the butler argument (B) from p. 2 is semantically valid.

E5.15. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Good translations.

b. Truth functional operators

c. Parse trees, interpretation functions and parallel trees

5.3 Quantificational

It is not surprising that our goals for the quantificational case remain very much as in the sentential one. We still want to produce translations — consisting of interpretation functions and formal sentences — which capture available structure, making a formal $\mathscr{P}'$ true at intended interpretation I_{ω} just when the corresponding ordinary $\mathscr{P}$ is true at story ω. We do this as before, by assuring that the various parts of the ordinary and formal languages work the same way. Of course, now we are interested in capturing *quantificational* structure, and the interpretation and formal sentences are for *quantificational* languages.

In the last section, we developed a recipe for translating from ordinary language into sentential expressions, associating particular bits or ordinary language with various formal symbols. We might proceed in very much the same way here, moving from our notion of *truth-functional* operators, to that of *extensional* terms, relation symbols, and operators. Roughly, an ordinary term is *extensional* when the truth value of a sentence in which it appears depends just on the object to which it refers; an ordinary relation symbol is *extensional* when the truth value of a sentence in which it appears depends just on the objects to which it applies; and an ordinary operator is *extensional* when the truth value of a sentence in which it appears depends just on the satisfaction of expressions which appear in its blanks. Clearly the notion of an extensional operator at least is closely related to that of a truth functional operator. Extensional terms, relation symbols and operators in ordinary language work very much like corresponding ones in a formal quantificational language — where, again, the idea would be to identify bits of ordinary language which contribute to truth values in the same way as corresponding parts of the formal language.

However, in the quantificational case, an official *recipe* for translation is relatively complicated. It is better to work directly with the fundamental goal of producing formal translations that are true in the same situations as ordinary expressions. To be sure, certain patterns and strategies will emerge, but, again, we should think of what we are doing less as applying a recipe, than as directly using our understanding of what makes ordinary and formal sentences true to produce good translations. With this in mind, let us move directly to sample cases, beginning with those that are relatively simple, and advancing to ones that are more complex.

5.3.1 Simple Quantifications

First, sentences without quantifiers work very much as in the sentential case. Consider a simple example. Say we are confronted with 'Bob is happy'. We might begin,

as in the sentential case, with the interpretation function,

B: Bob is happy

and use B for 'Bob is happy', $\sim B$ for 'Bob is not happy', and so forth. But this is to ignore structure we are now capable of capturing. Thus, in our standard quantificational language $\mathcal{L}_q$, we might let U be the set of all people, and set,

b: Bob

H^1: $\{\mathsf{o} \mid \mathsf{o}$ is a happy person$\}$

Then we can use Hb for 'Bob is happy', $\sim Hb$ for 'Bob is not happy', and so forth. If I_ω assigns Bob to b, and the set of happy things to H, then Hb is satisfied and true on I_ω just in case Bob is happy at ω — which is just what we want. Similarly suppose we are confronted with 'Bob's father is happy'. In the sentential case, we might have tried, F: Bob's father is happy. But this is to miss structure available to us now. So we might consider assigning a constant d to Bob's father and going with Hd as above. But this also misses available structure. In this case, we can expand the interpretation function to include,

f^1: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and n is the father of $\mathsf{m}\}$

Then for any variable assignment d, $\mathsf{I}_\mathsf{d}[b] =$ Bob and $\mathsf{I}_\mathsf{d}[f^1b]$ is Bob's father. So Hf^1b is satisfied and true just in case Bob's father is happy. $\sim Hf^1b$ is satisfied just in case Bob's father is not happy, and so forth — which is just what we want. In these cases without quantifiers, once we have translated simple sentences, everything else proceeds as in the sentential case. Thus, for example, for 'Neither Bob nor his father is happy' we might offer, $\sim Hb \wedge \sim Hf^1b$.

The situation gets more interesting when we add quantifiers. We will begin with cases where a quantifier's scope includes neither binary operators nor other quantifiers, and gradually increase complexity. Consider the following interpretation function.

I $\quad$ U: $\{\mathsf{o} \mid \mathsf{o}$ is a dog$\}$

f^1: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and n is the father of $\mathsf{m}\}$

W^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o will have its day$\}$

We assume that there is some definite content to a dog's having its day, and that every dog *has* a father — if a dog "Adam" has no father at all, we will not have specified a legitimate function. (Why?) Say we want to translate the following sentences.

(1) Every dog will have its day

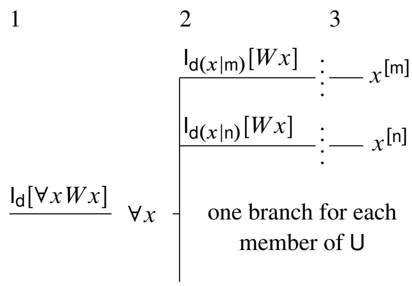
(2) Some dog will have its day

(3) Some dog will not have its day

(4) No dog will have its day

Assume 'some' means 'at least one'. The first sentence is straightforward. $\forall x Wx$ is read, 'for any x, Wx'; it is true just in case every dog will have its day. Suppose $\mathsf{I}\mathsf{I}_\omega$ is an interpretation I where the elements of U are m, n, and so forth. Then the tree is as below.

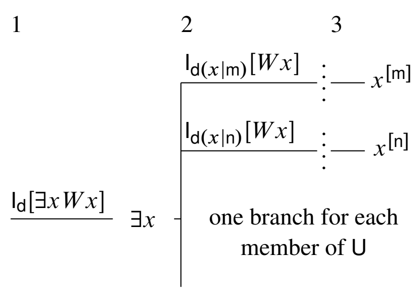
(AG)



The formula at (1) is satisfied just in case each of the branches at (2) is satisfied. But this can be the case only of each member of U is in the interpretation of W — which given our interpretation function, can only be the case if each dog will have its day. If even one dog does not have its day, then $\forall x Wx$ is not satisfied, and is not true.

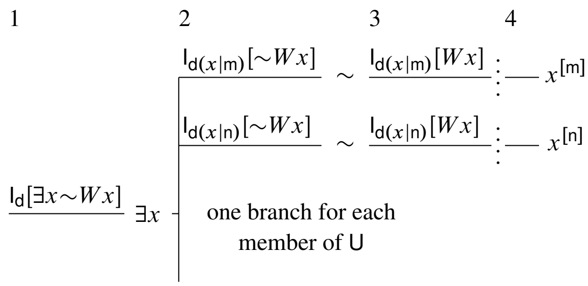
The second case is also straightforward. $\exists x Wx$ is read, 'there is an x such that Wx'; it is true just in case some dog will have its day.

(AH)



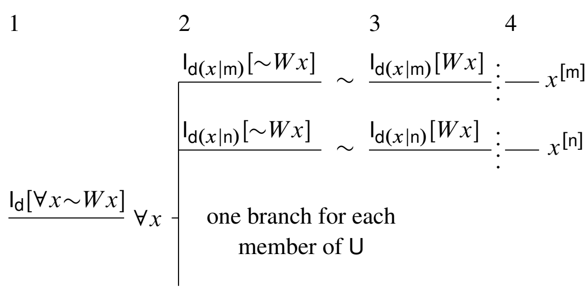
The formula at (1) is satisfied just in case at least one of the branches at (2) is satisfied. But this can be the case only of some member of U is in the interpretation of W — which, given the interpretation function, is to say that some dog will have its day.

The next two cases are only slightly more difficult. $\exists x \sim Wx$ is read, 'there is an x such that not Wx'; it is true just in case some dog will not have its day.

(AI) 

The formula at (1) is satisfied just in case at least one of the branches at (2) is satisfied. And a branch at (2) is satisfied just in case the corresponding branch at (3) is not satisfied. So $\exists x\sim Wx$ is satisfied and true just in case some member of U is not in the interpretation of W — just in case some dog does not have its day.

The last case is similar. $\forall x\sim Wx$ is read, 'for any x, not Wx'; it is true just in case every dog does not have its day.

(AJ) 

The formula at (1) is satisfied just in case all of the branches at (2) are satisfied. And this is the case just in case none of the branches at (3) are satisfied. So $\forall x\sim Wx$ is satisfied and true just in case none of the members of U are in the interpretation of W — just in case no dog has its day.

Perhaps it has already occurred to you that there are other ways to translate these sentences. The following lists what we have done, with "quantifier switching" alternatives on the right.

(AK)

Every dog will have its day	$\forall xWx$	$\sim\exists x\sim Wx$
Some dog will have its day	$\exists xWx$	$\sim\forall x\sim Wx$
Some dog will not have its day	$\exists x\sim Wx$	$\sim\forall xWx$
No dog will have its day	$\forall x\sim Wx$	$\sim\exists xWx$

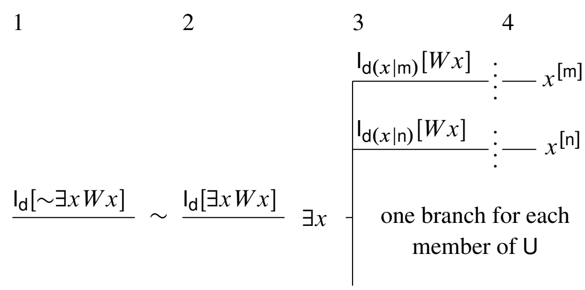
There are different ways to think about these alternatives. First, in ordinary language, beginning from the bottom, no dog will have its day, just in case not even one dog does. Similarly, moving up the list, some dog will not have its day, just in case not every dog does. And some dog will have its day just in case not every dog does not.

And every dog will have its day iff not even one dog does not. These equivalences may be difficult to absorb at first but, if you think about them, each should make sense.

Next, we might think about the alternatives purely in terms of abbreviations. Notice that, in a tree, $\mathsf{I_d}[\sim\sim\mathcal{P}]$ is always the same as $\mathsf{I_d}[\mathcal{P}]$ — the tildes "cancel each other out." But then, in the first case, $\sim\exists x\sim Wx$ abbreviates $\sim\sim\forall x\sim\sim Wx$ which is satisfied just in case $\forall xWx$ is satisfied. In the second case, $\exists xWx$ directly abbreviates $\sim\forall x\sim Wx$. In the third, $\exists x\sim Wx$ abbreviates $\sim\forall x\sim\sim Wx$ which is satisfied just in case $\sim\forall xWx$ is satisfied. And, in the last case, $\sim\exists xWx$ abbreviates $\sim\sim\forall x\sim Wx$, which is satisfied just in case $\forall x\sim Wx$ is satisfied. So, again, the alternatives are true under just the same conditions.

Finally, we might think about the alternatives directly, based on their branch conditions. Taking just the last case,

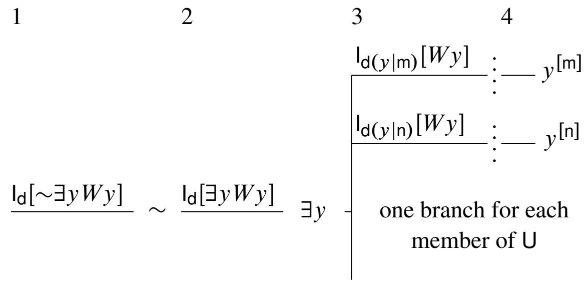
(AL)



The formula at (1) is satisfied just in case the formula at (2) is not. But the formula at (2) is not satisfied just in case none of the branches at (3) is satisfied — and this can only happen if no dog is in the interpretation of W, where this is as it should be for 'no dog will have its day'. In practice, there is no reason to prefer $\exists x\sim\mathcal{P}$ over $\sim\forall x\mathcal{P}$ or to prefer $\forall x\sim\mathcal{P}$ over $\sim\exists x\mathcal{P}$ — the choice is purely a matter of taste. It would be less natural to use $\sim\exists x\sim\mathcal{P}$ in place of $\forall x\mathcal{P}$, or $\sim\forall x\sim\mathcal{P}$ in place of $\exists x\mathcal{P}$. And it is a matter of good form to pursue translations that are natural. At any rate, all of the options satisfy CG. (But notice that we leave further room for alternatives among good answers, thus complicating comparisons with, for example, the back of the book!)

Observe that variables are *mere placeholders* for these expressions so that choice of variables also does not matter. Thus, in tree (AL) immediately above, the formula is true just in case no dog is in the interpretation of W. But we get the exact same result if the variable is y.

(AM)



In either case, what matters in the end is whether the objects are in the interpretation of the relation symbol: whether $\mathsf{m} \in \mathsf{I}[W]$, and so forth. If none are, then the formulas are satisfied. Thus the formulas are satisfied under *exactly* the same conditions. And since one is satisfied iff the other is satisfied, one is a good translation iff the other is. So the choice of variables is up to you.

Given all this, we continue to treat truth functional operators as before — and we can continue to use underlines to expose truth functional structure. The difference is that what we would have seen as "simple" sentences have structure we were not able to expose before. So, for example, 'Either every dog will have his day or no dog will have his day' gets translation, $\forall xWx \vee \forall x{\sim}Wx$; 'Some dog will have its day and some dog will not have its day', gets, $\exists xWx \wedge \exists x{\sim}Wx$; and so forth. If we want to say that some dog is such that its father will have his day, we might try $\exists xWf^1x$ — there is an x such that the *father of it* will have its day.

E5.16. On p. 174 we say that we may show directly, based on branch conditions, that the alternatives of table (AK) have the same truth conditions, but show it only for the last case. Use trees to demonstrate that the other alternatives are true under the same conditions. Be sure to explain how your trees have the desired results.

E5.17. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following. Assume Phil 300 is a logic class with Ninfa and Harold as members in which each student is associated with a unique homework partner.

U: $\{\mathsf{o} \mid \mathsf{o}$ is a student in Phil 300$\}$

a: Ninfa

d: Harold

p^1: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and n is the homework partner of $\mathsf{m}\}$

G^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o gets a good grade$\}$

H^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m gets a higher grade than $\mathsf{n}\}$

a. Ninfa and Harold both get a good grade.

b. Ninfa gets a good grade, but her homework partner does not.

c. Ninfa gets a good grade only if both her homework partner and Harold do.

d. Harold gets a higher grade than Ninfa.

*e. If Harold gets a higher grade than Ninfa, then he gets a higher grade than her homework partner.

f. Nobody gets a good grade.

*g. If someone gets a good grade, then Ninfa's homework partner does.

h. If Ninfa does not get a good grade, then nobody does.

*i. Nobody gets a grade higher than their own grade.

j. If no one gets a higher grade than Harold, then no one gets a good grade.

E5.18. Produce a good quantificational translation for each of the following. In this case you should provide an interpretation function for the sentences. Let U be the set of famous philosophers, and, assuming that each has a unique successor, implement a *successor* function.

a. Plato is a good philosopher.

b. Plato is better than Aristotle.

c. Neither Plato is better than Aristotle, nor Aristotle is better than Plato.

*d. If Plato is good, then his successor and successor's successor are good.

e. No philosopher is better than his successor.

f. Not every philosopher is better than Plato.

g. If all philosophers are good, then Plato and Aristotle are good.

h. If neither Plato nor his successor are good, then no philosopher is good.

*i. If some philosopher is better than Plato, then Aristotle is.

j. If every philosopher is better than his successor, then no philosopher is better than Plato.

5.3.2 Complex Quantifications

With a small change to our interpretation function, we introduce a new sort of complexity into our translations. Suppose U includes not just all dogs, but all physical objects, so that our interpretation function II has,

$$\begin{array}{lrl} \mathsf{II} & \mathsf{U}: & \{\mathsf{o} \mid \mathsf{o} \text{ is a physical object}\} \\ & W^1: & \{\mathsf{o} \mid \mathsf{o} \in \mathsf{U} \text{ and } \mathsf{o} \text{ will have its day}\} \\ & D^1: & \{\mathsf{o} \mid \mathsf{o} \in \mathsf{U} \text{ and } \mathsf{o} \text{ is a dog}\} \end{array}$$

Thus the universe includes more than dogs, and D is a relation symbol with application to dogs. We set out to translate the same sentences as before.[6]

(1) Every dog will have its day

(2) Some dog will have its day

(3) Some dog will not have its day

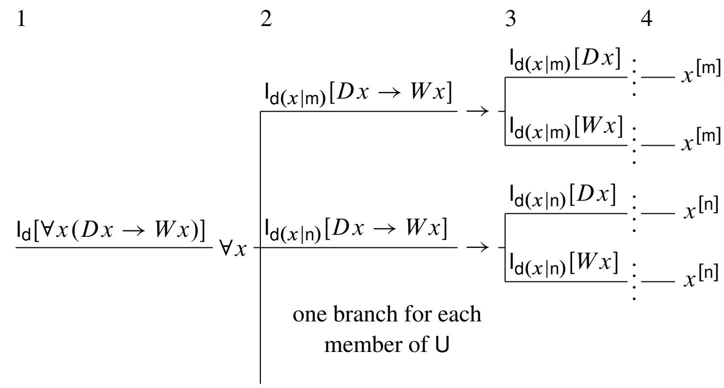
(4) No dog will have its day

This time, $\forall x Wx$ does *not* say that every dog will have its day. $\forall x Wx$ is true just in case everything in U, dogs along with everything else, will have its day. So it might be that every *dog* will have its day even though something else, for example my left sock, does not. So $\forall x Wx$ is not a good translation of 'every dog will have its day'.

We do better with $\forall x(Dx \to Wx)$. $\forall x(Dx \to Wx)$ is read, 'for any x if x is a dog, then x will have its day'; it is true just in case every dog will have its day. Again, suppose II_ω is an interpretation I such that the elements of U are m, n ....

[6] Sentences of the sort, 'all $\mathcal{P}$ are $\mathcal{Q}$', 'no $\mathcal{P}$ are $\mathcal{Q}$', 'some $\mathcal{P}$ are $\mathcal{Q}$', and 'some $\mathcal{P}$ are not $\mathcal{Q}$' are, in a tradition reaching back to Aristotle, often associated with a "square of opposition" and called A, E, I and O sentences. In a context with the full flexibility of quantifier languages, there is little point to the special treatment, insofar as our methods apply to these as well as to ones that are more complex. For discussion, see Pietroski, "Logical Form."

(AN)

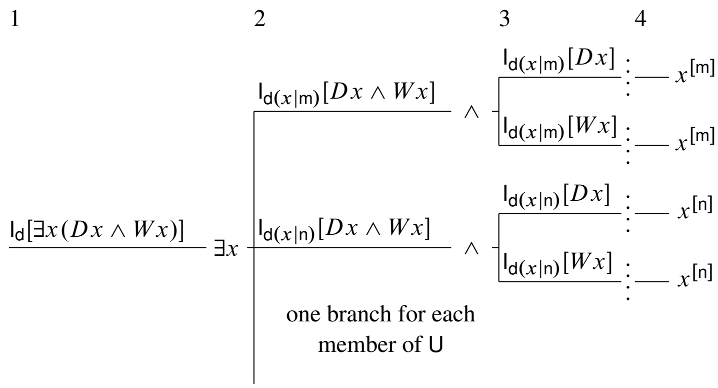


The formula at (1) is satisfied just in case each of the branches at (2) is satisfied. And all the branches at (2) are satisfied just in case there is no S/N pair at (3). This is so just in case nothing in U is a dog that does not have its day; that is, just in case every dog has its day. It is important to see how this works: There is a branch at (2) for *each* thing in U. The key is that branches for things that are not dogs are "vacuously" satisfied *just because the things are not dogs*. If $\forall x(Dx \to Wx)$ is true, however, whenever a branch is for a thing that is a dog — so that a top branch of a pair at (3) is satisfied, that thing must be one that will have its day. If anything is a dog that does not have its day, there is a S/N pair at (3), and $\forall x(Dx \to Wx)$ is not satisfied and not true.

It is worth noting some expressions that do not result in a good translation. $\forall xDx \wedge \forall xWx$ is true just in case everything is a dog and everything will have its day. To make it false, all it takes is one thing that is not a dog, or one thing that will not have its day — but this is not what we want. If this is not clear, work it out on a tree. Similarly, $\forall xDx \to \forall xWx$ is true just in case *if* everything is a dog, then everything will have its day. To make it true, all it takes is one thing that is not a dog — then the antecedent is false, and the conditional is true; but again, this is not what we want. In the good translation, $\forall x(Dx \to Wx)$, the quantifier picks out each thing in U, the antecedent of the conditional identifies the ones we want to talk about, and the consequent says what we want to say about them.

Moving on to the second sentence, $\exists x(Dx \wedge Wx)$ is read, 'there is an x such that x is a dog, and x will have its day'; it is true just in case some dog will have its day.

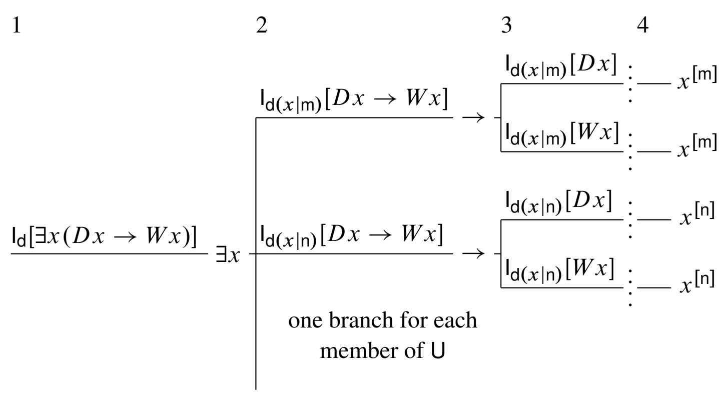
(AO)



The formula at (1) is satisfied just in case one of the branches at (2) is satisfied. A branch at (2) is satisfied just in cases both branches in the corresponding pair at (3) are satisfied. And this is so just in case something is a dog that will have its day.

Again, it is worth noting expressions that do not result in good translation. $\exists x Dx \wedge \exists x Wx$ is true just in case something is a dog, and something will have its day — where these need not be the same; so $\exists x Dx \wedge \exists x Wx$ might be true even though no *dog* has its day. $\exists x(Dx \rightarrow Wx)$ is true just in case something is such that *if* it is a dog, then it will have its day.

(AP)



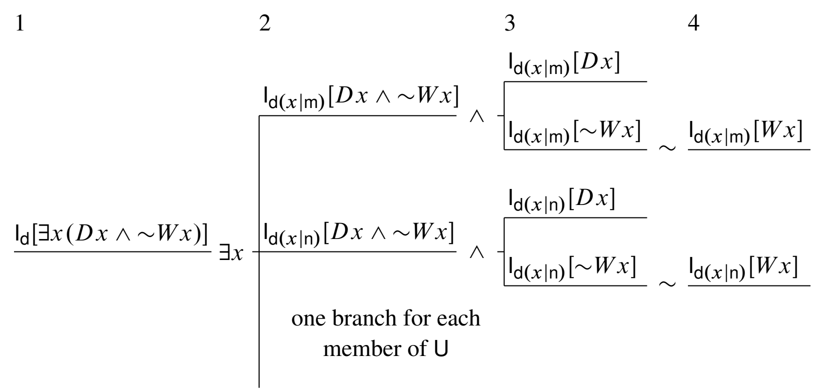
The formula at (1) is satisfied just in case one of the branches at (2) is satisfied; and a branch at (2) is satisfied just in case there is a pair at (3) in which the top is N or the bottom is S. So all we need for $\exists x(Dx \rightarrow Wx)$ to be true is for there to be even one thing that is not a dog — for example, my sock — or one thing that will have its day. So $\exists x(Dx \rightarrow Wx)$ can be true though no dog has its day.

The cases we have just seen are typical. Ordinarily, the existential quantifier operates on expressions with main operator $\wedge$. If it operates on an expression with main operator $\rightarrow$, the resultant expression is satisfied just by virtue of something

that does not satisfy the antecedent. And, ordinarily, the universal quantifier operates on expressions with main operator $\rightarrow$. If it operates on an expression with main operator $\wedge$, the expression is satisfied only if *everything* in U has features from both parts of the conjunction — and it is uncommon to say something about everything in U, as opposed to all the objects of a certain sort. Again, when the universal quantifier operates on an expression with main operator $\rightarrow$, the antecedent of the conditional identifies the objects we want to talk about, and the consequent says what we want to say about them.

Once we understand these two cases, the next two are relatively straightforward. $\exists x(Dx \wedge \sim Wx)$ is read, 'there is an x such that x is a dog and x will not have its day'; it is true just in case some dog will not have its day. Here is the tree without branches for the (by now obvious) term assignments.

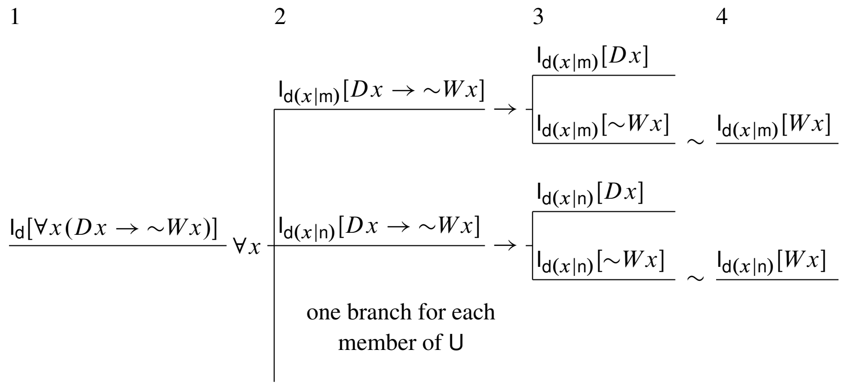
(AQ)



The formula at (1) is satisfied just in case some branch at (2) is satisfied. A branch at (2) is satisfied just in case the corresponding pair of branches at (3) is satisfied. And for a lower branch at (3) to be satisfied, the corresponding branch at (4) has to be unsatisfied. So for $\exists x(Dx \wedge \sim Wx)$ to be satisfied, there has to be something that is a dog and does not have its day. In principle, this is just like, 'some dog will have its day'. We set out to say that some object of sort $\mathcal{P}$ has feature $\mathcal{Q}$. For this, we say that there is an x hat is of type $\mathcal{P}$, and has feature $\mathcal{Q}$. In 'some dog will have its day', $\mathcal{Q}$ is the simple W. In this case, $\mathcal{Q}$ is the slightly more complex $\sim W$.

Finally, $\forall x(Dx \rightarrow \sim Wx)$ is read, 'for any x, if x is a dog, then x will not have its day'; it is true just in case every dog will not have its day — that is, just in case no dog will have its day.

(AR)



The formula at (1) is satisfied just in case every branch at (2) is satisfied. Every branch at (2) is satisfied just in case there is no S/N pair at (3); and for this to be so there cannot be a case where a top at (3) is satisfied, and the corresponding bottom at (4) is satisfied as well. So $\forall x(Dx \rightarrow \sim Wx)$ is satisfied and true just in case nothing is a dog that will have its day. Again, in principle, this is like 'every dog will have its day'. Using the universal quantifier, we pick out the class of things we want to talk about in the antecedent, and say what we want to say about the members of the class in the consequent. In this case, what we want to say is that things in the class will not have their day.

As before, quantifier-switching alternatives are possible. In the table below, alternatives to what we have done are listed on the right.

(AS)

Every dog will have its day	$\forall x(Dx \rightarrow Wx)$	$\sim\exists x(Dx \wedge \sim Wx)$
Some dog will have its day	$\exists x(Dx \wedge Wx)$	$\sim\forall x(Dx \rightarrow \sim Wx)$
Some dog will not have its day	$\exists x(Dx \wedge \sim Wx)$	$\sim\forall x(Dx \rightarrow Wx)$
No dog will have its day	$\forall x(Dx \rightarrow \sim Wx)$	$\sim\exists x(Dx \wedge Wx)$

Beginning from the bottom, if not even one thing is a dog that will have its day, then no dog will have its day. Moving up, if it is not the case that everything that is a dog will have its day, then some dog will not. Similarly, if it is not the case that everything that is a dog will not have its day, then some dog does. And if not even one thing is a dog that does not have its day, then every dog will have its day. Again, choices among the alternatives are a matter of taste, though the latter ones may be more natural than the former. If you have any questions about how the alternatives work, work them through on trees.

Before turning to some exercises, let us generalize what we have done a bit. Include in our interpretation function,

H^1: {o | o is happy}

C^1: {o | o is a cat}

Suppose we want to say, not that every dog will have its day, but that every happy dog will have its day. Again, in principle this is like what we have done. With the universal quantifier, we pick out the class of things we want to talk about in the antecedent — in this case, happy dogs, and say what we want about them in the consequent. Thus $\forall x[(Dx \wedge Hx) \rightarrow Wx]$ is true just in case everything that is both happy and a dog will have its day, which is to say, every happy dog will have its day. Similarly, if we want to say, every dog will or will not have its day, we might try, $\forall x[Dx \rightarrow (Wx \vee \sim Wx)]$. Or putting these together, for 'every happy dog will or will not have its day', $\forall x[(Dx \wedge Hx) \rightarrow (Wx \vee \sim Wx)]$. We consistently pick out the things we want to talk about in the antecedent, and say what we want about them with the consequent. Similar points apply to the existential quantifier. Thus 'Some happy dog will have its day' has natural translation, $\exists x[(Dx \wedge Hx) \wedge Wx]$ — something is a happy dog and will have its day. 'Some happy dog will or will not have its day' gets, $\exists x[(Dx \wedge Hx) \wedge (Wx \vee \sim Wx)]$. And so forth.

It is tempting to treat, 'All dogs and cats will have their day' similarly with translation, $\forall x[(Dx \wedge Cx) \rightarrow Wx]$. But this would be a mistake! We do not want to say that everything which is a dog *and* a cat will have its day — for nothing is both a dog and a cat! Rather, good translations are, $\forall x(Dx \rightarrow Wx) \wedge \forall x(Cx \rightarrow Wx)$ — all dogs will have their day *and* all cats will have their day or the more elegant, $\forall x[(Dx \vee Cx) \rightarrow Wx]$ — each thing that is either a dog *or* a cat will have its day. In the happy dog case, we needed to *restrict* to class under consideration to include just happy dogs; in this dog and cat case, we are not restricting the class, but rather expanding it to include both dogs and cats. The disjunction $(Dx \vee Cx)$ applies to things in the broader class which includes both dogs and cats.

This dog and cat case brings out the point that we do not merely "cookbook" from ordinary language to formal translations, but rather want truth conditions to match. And we can make the conditions match for expressions where standard language does not lie directly on the surface. Thus, consider, 'Only dogs will have their day'. This does *not* say that all dogs will have their day. Rather it tells us that if something has its day, then it is a dog, $\forall x(Wx \rightarrow Dx)$. Similarly, 'No dogs, except the happy ones, will have their day', tells us that dogs that are not happy will not have their day, $\forall x[(Dx \wedge \sim Hx) \rightarrow \sim Wx]$. It is tempting to add that the happy dogs will have their day, but it is not clear that this is part of what we have actually *said*; 'except' seems precisely to *except* members of the specified class from what is said.[7]

[7] It may be that we conventionally use 'except' in contexts where the consequent is reversed for the excepted class, for example, 'I like all foods except brussels sprouts' — where I say it this way *because*

Further, as in the dog and cat case, sometimes surface language is positively misleading compared to standard readings. Consider, for example, 'if some dog is happy, it will have its day', and 'if any dog is happy, then they all are'. It is tempting to translate the first, $\exists x[(Dx \wedge Hx) \rightarrow Wx]$ — but this is not right. All it takes to make this expression true is something that is not a happy dog (for example, my sock); if something is not a happy dog, then a branch for the conditional is satisfied, so that the existentially quantified expression is satisfied. But we want rather to say something about *all* dogs — if some (*arbitrary*) dog is happy it will have its day — so that no matter what dog you pick, if it is happy, then it will have its day; thus the correct translation is $\forall x[(Dx \wedge Hx) \rightarrow Wx]$. Similarly, it may be tempting to translate, the 'any' of 'if any dog is happy, then they all are' by the universal quantifier. But the correct translation is rather, $\exists x(Dx \wedge Hx) \rightarrow \forall x(Dx \rightarrow Hx)$ — if *some* dog is happy, then every dog is happy. The best way to approach these cases is to think directly about the *conditions* under which the ordinary expressions are true and false, and to produce formal translations that are true and false under the same conditions. For these last cases however, it is worth noting that when there is "pronominal" cross reference as, 'if some/any $\mathscr{P}$ is $\mathscr{Q}$ then *it* has such-and-such features' the statement translates most naturally with the universal quantifier. But when such cross-reference is absent as, 'if some/any $\mathscr{P}$ is $\mathscr{Q}$ then so-and-so is such-and-such' the statement translates naturally as a conditional with an existential antecedent. The point is not that there are no grammatical cues! But cues are not so simple that we can always simply read from 'some' to the existential quantifier, and from 'any' to the universal. Perhaps this is sufficient for us to move to the following exercises.

E5.19. Use trees to show that the quantifier-switching alternatives from (AS) are true and false under the same conditions as their counterparts. Be sure to explain how your trees have the desired results.

E5.20. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following. (Perhaps these sentences reflect residual frustration over a Mustang the author owned in graduate school).

U: $\{\mathsf{o} \mid \mathsf{o}$ is a car$\}$

T^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a Toyota$\}$

F^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a Ford$\}$

I do not like brussels sprouts. But, again, it is not clear that I have actually said whether I like them or not.

E^1: {o | o ∈ U and o was built in the eighties}
J^1: {o | o ∈ U and o is a piece of junk}
R^1: {o | o ∈ U and o is reliable}

a. Some Ford is a piece of junk.

*b. Some Ford is an unreliable piece of junk.

c. Some Ford built in the eighties is a piece of junk.

d. Some Ford built in the eighties is an unreliable piece of junk.

e. Any Ford is a piece of junk.

f. Any Ford is an unreliable piece of junk.

*g. Any Ford built in the eighties is a piece of junk.

h. Any Ford built in the eighties is an unreliable piece of junk.

i. No reliable car is a piece of junk.

j. No Toyota is an unreliable piece of junk.

*k. If a car is unreliable, then it is a piece of junk.

l. If some Toyota is unreliable, then every Ford is.

m. Only Toyotas are reliable.

n. Not all Toyotas and Fords are reliable.

o. Any car, except for a Ford, is reliable.

E5.21. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following. Assume that Bob is married, and that each married person has a unique "primary" spouse in case of more than one.

U: {o | o is a person who is married}
b: Bob
s^1: {⟨m, n⟩ | n is the (primary) spouse of m}

A^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is having an affair$\}$

E^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is employed$\}$

H^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is happy$\}$

L^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m loves $\mathsf{n}\}$

M^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}$ is married to $\mathsf{n}\}$

a. Bob's spouse is happy.

*b. Someone is married to Bob.

c. Anyone who loves their spouse is happy.

d. Nobody who is happy and loves their spouse is having an affair.

e. Someone is happy just in case they are employed.

f. Someone is happy just in case someone is employed.

g. Some happy people have affairs, and some do not.

*h. Anyone who loves and is loved by their spouse is happy, though some are not employed.

i. Only someone who loves their spouse and is employed is happy.

j. Anyone who is unemployed and whose spouse is having an affair is unhappy.

k. People who are unemployed and people whose spouse is having an affair are unhappy.

*l. Anyone married to Bob is happy if Bob is not having an affair.

m. Anyone married to Bob is happy only if Bob is employed and is not having an affair.

n. If Bob is having an affair, then everyone married to him is unhappy, and nobody married to him loves him.

o. Only unemployed people and unhappy people have affairs, but if someone loves and is loved by their spouse, then they are happy unless they are unemployed.

E5.22. Produce a good quantificational translation for each of the following. You should produce a single interpretation function with application to all of the sentences. Let U be the set of all animals.

a. Not all animals make good pets.

b. Dogs and cats make good pets.

c. Some dogs are ferocious and make good pets, but no cat is both.

d. No ferocious animal makes a good pet, unless it is a dog.

e. No ferocious animal makes a good pet, unless Lassie is both.

f. Some, but not all good pets are dogs.

g. Only dogs and cats make good pets.

h. Not all dogs and cats make good pets, but some do.

i. If Lassie does not make a good pet, then the only good pet is a cat that is ferocious, or a dog that is not.

j. A dog or cat makes a good pet if and only if it is not ferocious.

5.3.3 Overlapping Quantifiers

The full power of our quantificational languages emerges only when we allow one quantifier to appear in the scope of another.[8] So let us turn to some cases of this sort. First, let U be the set of all people, and suppose the intended interpretation of L^2 is $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$, and m loves $\mathsf{n}\}$. Say we want to translate,

(1) Everyone loves everyone.

(2) Someone loves someone.

(3) Everyone loves someone.

(4) Everyone is loved by someone.

(5) Someone loves everyone.

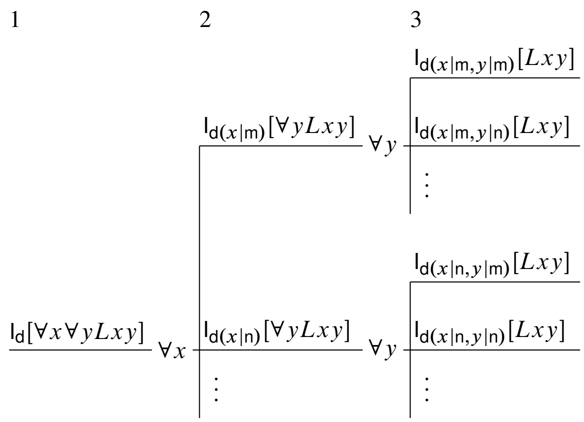
[8] Aristotle's categorical logic is capable of handling simple A, E, I, and O sentences — consider experience you may have had with "Venn diagrams." But you will not be able to make his logic, or such diagrams apply to the full range of cases that follow (see note 6)!

(6) Someone is loved by everyone.

First, you should be clear how each of these differs from the others. In particular, it is enough for (4) ‘everyone is loved by someone’ that for each person there is a lover of them — perhaps their mother (or themselves); but for (6) ‘someone is loved by everyone’ we need some one person, say Elvis, that everyone loves. Similarly, it is enough for (3) ‘everyone loves someone’ that each person loves some person — perhaps their mother (or themselves); but for (5) ‘someone loves everyone’ we need some particularly loving individual, say Mother Theresa, who loves everyone.

The first two are straightforward. $\forall x \forall y L x y$ is read, ‘for any x and any y, x loves y; it is true just in case everyone loves everyone.

(AT)

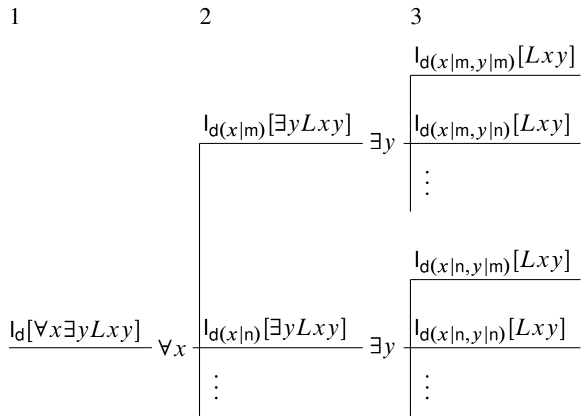


The branch at (1) is satisfied just in case all of the branches at (2) are satisfied. And all of the branches at (2) are satisfied just in case all of the branches at (3) are satisfied. But every combination of objects appears at the branch tips. So $\forall x \forall y L x y$ is satisfied and true just in case for any pair $\langle \mathsf{m}, \mathsf{n}\rangle \in \mathsf{U}^2$, $\langle \mathsf{m}, \mathsf{n}\rangle$ is in the interpretation of L. Notice that the order of the quantifiers and variables makes no difference: for a given interpretation I, $\forall x \forall y L y x$, $\forall y \forall x L x y$, and $\forall y \forall x L y x$ are all satisfied and true under the same condition — just when every $\langle \mathsf{m}, \mathsf{n}\rangle \in \mathsf{U}^2$ is a member of $\mathsf{I}[L]$.

The case for the second sentence is similar. $\exists x \exists y L x y$ is read, ‘there is an x and there is a y such that x loves y; it is true just in case some $\langle \mathsf{m}, \mathsf{n}\rangle \in \mathsf{U}^2$ is a member of $\mathsf{I}[L]$ — just in case someone loves someone. The tree is like (AT) above, but with $\exists$ uniformly substituted for $\forall$. Then the formula at (1) is satisfied iff a branch at (2) is satisfied; iff a branch at (3) is satisfied; iff someone loves someone. Again the order of the quantifiers does not matter.

The next cases are more interesting. $\forall x \exists y L x y$ is read, ‘for any x there is a y such that x loves y’; it is true just in case everyone loves someone.

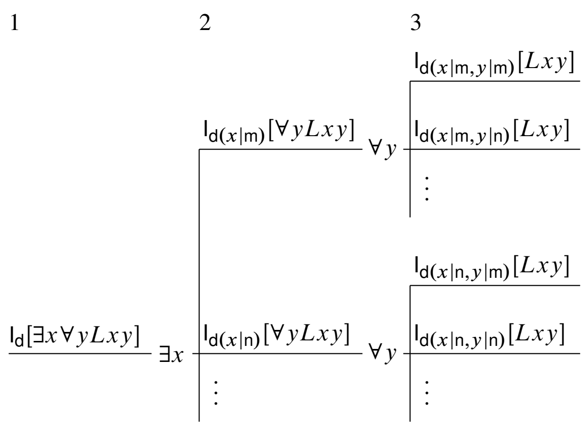
(AU)



The branch at (1) is satisfied just in case each of the branches at (2) is satisfied. And a branch at (2) is satisfied just in case at least one of the corresponding branches at (3) is satisfied. So $\forall x\exists yLxy$ is satisfied just in case, no matter which o you pick, there is some p such that such that o loves p — so that everyone loves someone. This time, the order of the of the variables makes a difference: thus, $\forall x\exists yLyx$ translates sentence (4). The picture is like the one above, with Lyx uniformly replacing Lxy. This expression is satisfied just in case no matter which o you pick, there is some p such that such that p loves o — so that everyone is loved by someone.

Finally, $\exists x\forall yLxy$ is read, 'there is an x such that for any y, x loves y'; it is satisfied and true just in case someone loves everyone.

(AV)



The branch at (1) is satisfied just in case some branch at (2) is satisfied. And a branch at (2) is satisfied just in case *each* of the corresponding branches at (3) is satisfied. So $\exists x\forall yLxy$ is satisfied and true just in case there is some $\mathsf{o} \in \mathsf{U}$ such that, no matter what $\mathsf{p} \in \mathsf{U}$ you pick, $\langle \mathsf{o}, \mathsf{p} \rangle \in \mathsf{I}[L]$ — just when there is someone who loves everyone. If we switch Lyx for Lxy, we get a tree for $\exists x\forall yLyx$; this formula is true

just when someone is loved by everyone. Switching the order of the quantifiers and variables makes no difference when quantifiers are the same. But it matters crucially when quantifiers are different!

Let us see what happens when, as before, we broaden the interpretation function so that U includes all physical objects.

II U: {o | o is a physical object}
P^1: {o | o $\in$ U and o is a person}
L^2: {$\langle$m, n$\rangle$ | m, n $\in$ U, and m loves n}

Let us set out to translate the same sentences as before.

For 'everyone loves everyone', where we are talking about *people*, $\forall x \forall y Lxy$ will not do. $\forall x \forall y Lxy$ requires that each member of U love all the other members of U — but then we are requiring that my left sock love my computer, and so forth. What we need is rather, $\forall x \forall y[(Px \wedge Py) \rightarrow Lxy]$. With the last branch tips omitted, the tree is as follows.

(AW)

1 $\mathsf{I_d}[\forall x \forall y((Px \wedge Py) \rightarrow Lxy)]$ $\forall x$

2 $\mathsf{I_{d(x|m)}}[\forall y((Px \wedge Py) \rightarrow Lxy)]$ $\forall y$

3 $\mathsf{I_{d(x|m,y|m)}}[(Px \wedge Py) \rightarrow Lxy]$ $\rightarrow$
4 $\mathsf{I_{d(x|m,y|m)}}[Px \wedge Py]$
4 $\mathsf{I_{d(x|m,y|m)}}[Lxy]$

3 $\mathsf{I_{d(x|m,y|n)}}[(Px \wedge Py) \rightarrow Lxy]$ $\rightarrow$
4 $\mathsf{I_{d(x|m,y|n)}}[Px \wedge Py]$
4 $\mathsf{I_{d(x|m,y|n)}}[Lxy]$

3 $\vdots$

2 $\mathsf{I_{d(x|n)}}[\forall y((Px \wedge Py) \rightarrow Lxy)]$ $\forall y$

3 $\mathsf{I_{d(x|n,y|m)}}[(Px \wedge Py) \rightarrow Lxy]$ $\rightarrow$
4 $\mathsf{I_{d(x|n,y|m)}}[Px \wedge Py]$
4 $\mathsf{I_{d(x|n,y|m)}}[Lxy]$

3 $\mathsf{I_{d(x|n,y|n)}}[(Px \wedge Py) \rightarrow Lxy]$ $\rightarrow$
4 $\mathsf{I_{d(x|n,y|n)}}[Px \wedge Py]$
4 $\mathsf{I_{d(x|n,y|n)}}[Lxy]$

3 $\vdots$

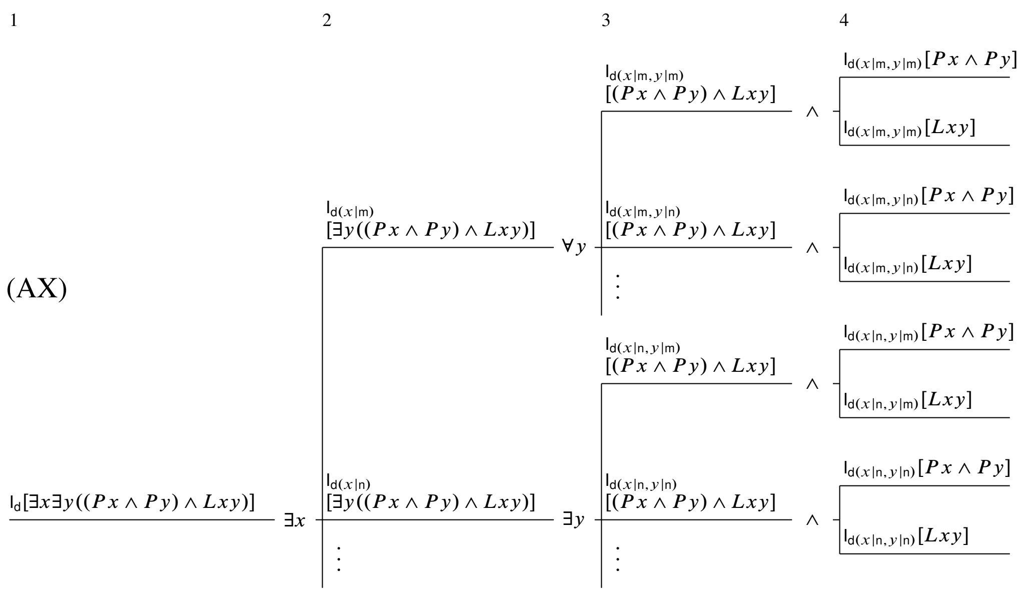
2 $\vdots$

The formula at (1) is satisfied iff all the branches at (2) are satisfied; all the branches at (2) are satisfied just in case all the branches at (3) are satisfied. And, for this to be the case, there can be no pair at (4) where the top is satisfied and the bottom is not. That is, there can be no o and p such that o and p are people, o, p $\in$ I[P], but o does not love p, $\langle$o, p$\rangle \notin$ I[L]. The idea is very much as before: With the universal

quantifiers, we select the things we want to talk about in the antecedent, we make sure that x and y pick out *people*, and then say what we want to say about the things in the consequent.

The case for 'someone loves someone' also works on close analogy with what has gone before. In this case, we do not use the conditional. If the quantifiers in the above tree were existential, all we would need is *one* branch at (2) to be satisfied, and *one* branch at (3) satisfied. And, for this, all we would need is one thing that is not a person — so that the top branch for the conditional is N, and the conditional is therefore S. On the analogy with what we have seen before, what we want is something like, $\exists x \exists y[(Px \wedge Py) \wedge Lxy]$. There are some *people* x and y such that x loves y.

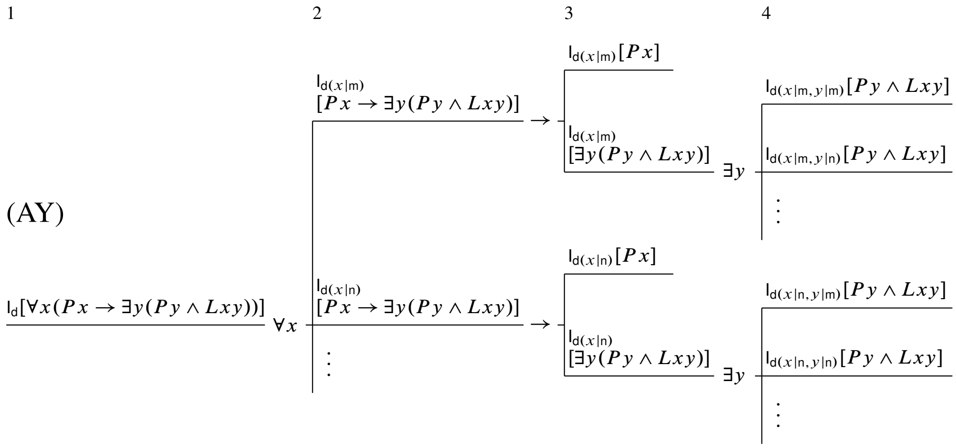
(AX)



The formula at (1) is satisfied iff at least one branch at (2) is satisfied. At least one branch at (2) is satisfied just in case at least one branch at (3) is satisfied. And for this to be the case, we need some branch pair at (4) where both the top and the bottom are satisfied — some o and p such that o and p are people, $\mathsf{o}, \mathsf{p} \in \mathsf{I}[P]$, and o loves p, $\langle \mathsf{o}, \mathsf{p} \rangle \in \mathsf{I}[L]$.

In these cases, the order of the quantifiers and variables does not matter. But order matters when quantifiers are mixed. Thus, for 'everyone loves someone', $\forall x[Px \rightarrow \exists y(Py \wedge Lxy)]$ is good — if any thing x is a person, then there is some y such that y is a person and x loves y.

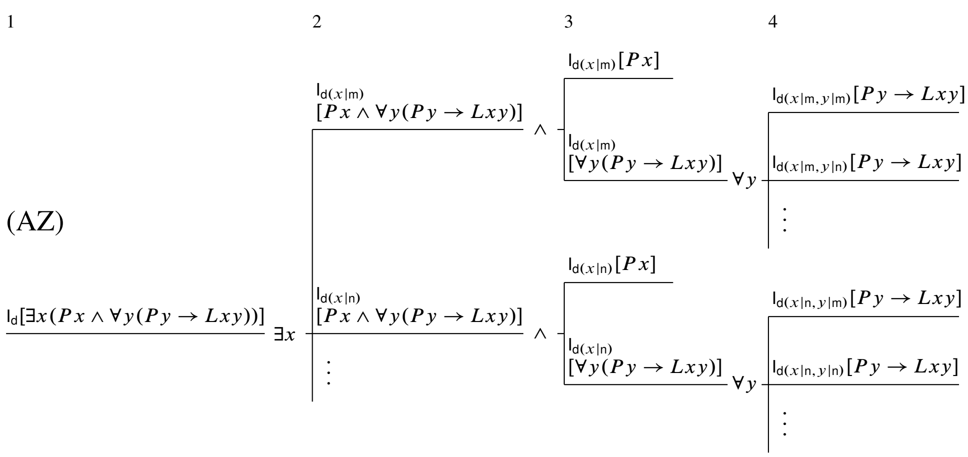
(AY)



The formula at (1) is satisfied just in case all the branches at (2) are satisfied. All the branches at (2) are satisfied just in case no pair at (3) has the top satisfied and the bottom not. If x is assigned to something that is not a person, the branch at (2) is satisfied trivially. But where the assignment to x is some o that is a person, a bottom branch at (3) is satisfied just in case at least one of the corresponding branches at (4) is satisfied — just in case there is some p such that p is a person and o loves p. Notice, again, that the universal quantifier is associated with a conditional, and the existential with a conjunction. Similarly, we translate 'everyone is loved by someone, $\forall x[Px \to \exists y(Py \wedge Lyx)]$. The tree is as above, with Lxy uniformly replaced by Lyx.

For 'someone loves everyone, $\exists x[Px \wedge \forall y(Py \to Lxy)]$ is good — there is an x such that x is a person, and for any y, if y is a person, then x loves y.

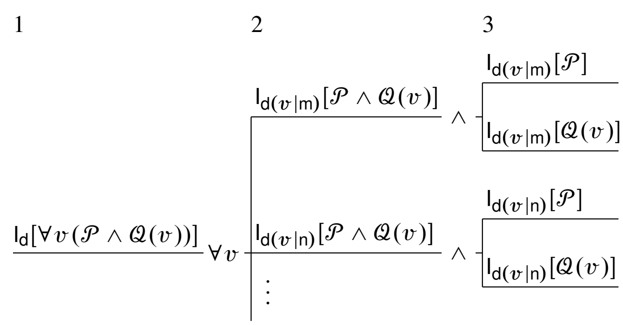
(AZ)



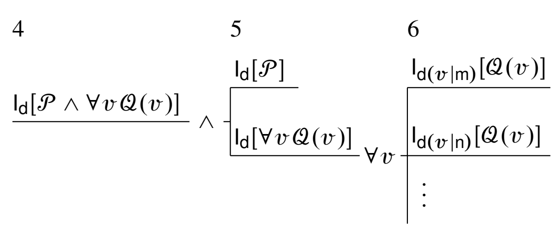
The formula at (1) is satisfied just in case some branch at (2) is satisfied. A branch at

(2) is satisfied just in case the corresponding pair at (3) is satisfied. The top of such a pair is satisfied when the assignment to x is some $\mathsf{o} \in \mathsf{I}[P]$; the bottom is satisfied just in case all of the corresponding branches at (4) are satisfied — just in case any p is such that *if* it is a person, then o loves it. So there has to be an o that loves every p. Similarly, you should be able to see that $\exists x[Px \wedge \forall y(Py \rightarrow Lyx)]$ is good for 'someone is loved by everyone'.

Again, it may have occurred to you already that there are other options for these sentences. This time natural alternatives are not for quantifier switching, but for quantifier *placement*. For 'someone loves everyone' we have given, $\exists x[Px \wedge \forall y(Py \rightarrow Lxy)]$ with the universal quantifier on the inside. However, $\exists x \forall y[Px \wedge (Py \rightarrow Lxy)]$ would do as well. As a matter of strategy, it may be best to keep quantifiers as close as possible to that which they modify. However, we can show that, in this case, pushing the quantifier across that which it does not bind leaves the truth condition unchanged. Let us make the point generally. Say $\mathcal{Q}(v)$ is a formula with variable v free, but $\mathcal{P}$ is one in which v is not free. We are interested in the relation between $(\mathcal{P} \wedge \forall v \mathcal{Q}(v))$ and $\forall v(\mathcal{P} \wedge \mathcal{Q}(v))$. Here are the trees.

(BA)


and,

(BB)


The key is this: Since $\mathcal{P}$ has no free instances of v, for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}}[\mathcal{P}]$ is satisfied just in case $\mathsf{I}_{\mathsf{d}(v|\mathsf{o})}[\mathcal{P}]$ is satisfied; for if v is not free in $\mathcal{P}$, the assignment to v makes no difference to the evaluation of $\mathcal{P}$. In (BA), the formula at (1) is satisfied iff each of the branches at (2) is satisfied; and each of the branches at (2) is satisfied iff each of the branches at (3) is satisfied. In (BB) the formula at (4) is satisfied iff both branches

at (5) are satisfied. The bottom requires that all the branches at (6) are satisfied. But the branches at (6) are just like the bottom branches from (3) in (BA). And given the equivalence between $\mathsf{I}_{\mathsf{d}}[\mathscr{P}]$ and $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathscr{P}]$, the top at (5) is satisfied iff each of the tops at (3) is satisfied. So the one formula is satisfied iff the other is as well. Notice that this only works because v is not free in $\mathscr{P}$. So you can move the quantifier past the $\mathscr{P}$ only if it does not bind a variable free in $\mathscr{P}$!

Parallel reasoning would work for any combination of $\forall$ and $\exists$, with $\land$, $\lor$ and $\to$. That is, supposing that v is not free in $\mathscr{P}$, each of the following pairs is equivalent.

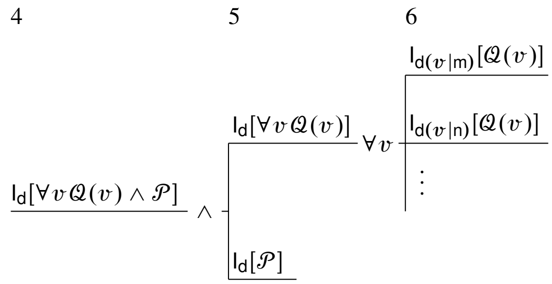
(BC)
$$
\begin{array}{lcl}
\forall v(\mathscr{P} \land \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \land \forall v\mathscr{Q}(v)\\
\exists v(\mathscr{P} \land \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \land \exists v\mathscr{Q}(v)\\
\forall v(\mathscr{P} \lor \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \lor \forall v\mathscr{Q}(v)\\
\exists v(\mathscr{P} \lor \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \lor \exists v\mathscr{Q}(v)\\
\forall v(\mathscr{P} \to \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \to \forall v\mathscr{Q}(v)\\
\exists v(\mathscr{P} \to \mathscr{Q}(v)) & \Longleftrightarrow & \mathscr{P} \to \exists v\mathscr{Q}(v)
\end{array}
$$

The comparison between $\forall y[Px \land (Py \to Lxy)]$ and $[Px \land \forall y(Py \to Lxy)]$ is an instance of the first pair. In effect, then, we can "push" the quantifier into the parentheses across a formula to which the quantifier does not apply, and "pull" it out across a formula to which the quantifier does not apply — without changing the conditions under which the formula is satisfied.

But we need to be more careful when the order of $\mathscr{P}$ and $\mathscr{Q}(v)$ is reversed. Some cases work the way we expect. Consider $\forall v(\mathscr{Q}(v) \land \mathscr{P})$ and $(\forall v\mathscr{Q}(v) \land \mathscr{P})$.

(BD)
$$
\begin{array}{llll}
1 & 2 & & 3\\
 & & & \mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathscr{Q}(v)]\\
 & \mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathscr{Q}(v) \land \mathscr{P}] & \land & \\
 & & & \mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathscr{P}]\\
 & & & \mathsf{I}_{\mathsf{d}(v|\mathsf{n})}[\mathscr{Q}(v)]\\
\mathsf{I}_{\mathsf{d}}[\forall v(\mathscr{Q}(v) \land \mathscr{P})] \;\; \forall v & \mathsf{I}_{\mathsf{d}(v|\mathsf{n})}[\mathscr{Q}(v) \land \mathscr{P}] & \land & \\
 & & & \mathsf{I}_{\mathsf{d}(v|\mathsf{n})}[\mathscr{P}]\\
 & \vdots & &
\end{array}
$$

and,

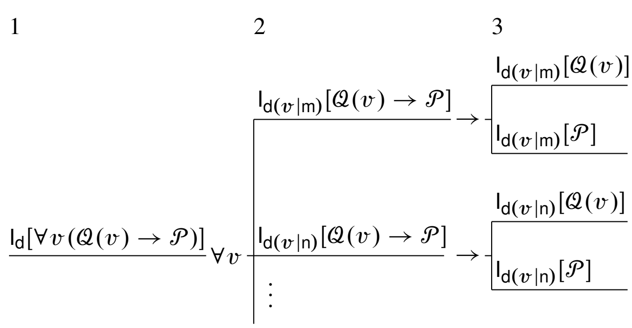
(BE) 

In this case, the reasoning is as before. In (BD), the formula at (1) is satisfied iff all the branches at (2) are satisfied; and all the branches at (2) are satisfied iff all the branches at (3) are satisfied. And in (BE), the formula at (4) is satisfied iff both branches at (5) are satisfied. And the top at (5) is satisfied iff all the branches at (6) are satisfied. But the branches at (6) are like the tops at (3). And given the equivalence between $\mathsf{I}_{\mathsf{d}}[\mathcal{P}]$ and $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}]$, the bottom at (5) is satisfied iff the bottoms at (3) are satisfied. So, again, the formulas are satisfied under the same conditions. And similarly for different combinations of the quantifiers $\forall$ or $\exists$ and the operators $\wedge$ or $\vee$. Thus our table extends as follows.

(BF)
$$\begin{array}{lcl}
\forall v\,(\mathcal{Q}(v) \wedge \mathcal{P}) & \Longleftrightarrow & (\forall v\,\mathcal{Q}(v) \wedge \mathcal{P}) \\
\exists v\,(\mathcal{Q}(v) \wedge \mathcal{P}) & \Longleftrightarrow & (\exists v\,\mathcal{Q}(v) \wedge \mathcal{P}) \\
\forall v\,(\mathcal{Q}(v) \vee \mathcal{P}) & \Longleftrightarrow & (\forall v\,\mathcal{Q}(v) \vee \mathcal{P}) \\
\exists v\,(\mathcal{Q}(v) \vee \mathcal{P}) & \Longleftrightarrow & (\exists v\,\mathcal{Q}(v) \vee \mathcal{P})
\end{array}$$

We can push a quantifier "into" the front part of a parenthesis or pull it out as above.

But the case is different when the main operator is $\rightarrow$. Consider trees for $\forall v\,(\mathcal{Q}(v) \rightarrow \mathcal{P})$ and, noting the quantifier shift, for $(\exists v\,\mathcal{Q}(v) \rightarrow \mathcal{P})$.

(BG) 

and

(BH)

$$\begin{array}{llll}
4 & 5 & & 6 \\
 & & & \mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathcal{Q}(v)] \\
 & \mathsf{I}_{\mathsf{d}}[\exists v\, \mathcal{Q}(v)] & \exists v & \mathsf{I}_{\mathsf{d}(v|\mathsf{n})}[\mathcal{Q}(v)] \\
 & & & \vdots \\
\mathsf{I}_{\mathsf{d}}[\exists v\, \mathcal{Q}(v) \to \mathcal{P}] \quad \to & & & \\
 & \mathsf{I}_{\mathsf{d}}[\mathcal{P}] & &
\end{array}$$

The formula at (4) is satisfied so long as at (5) the upper branch is N or bottom is S; and the top is N iff no branch at (6) is S; thus the formula at (4) is satisfied so long as none of the branches at (6) are S or the bottom at (5) is S; or, put the other way around, the formula at (4) is N iff one of the branches at (6) is S and the bottom at (5) is N. The formula at (1) is satisfied iff all the branches at (2) are satisfied; and all the branches at (2) are satisfied iff there is no S/N pair at (3); so the formula at (1) is N iff there is an S/N pair at (3). But, as before, the tops at (3) are the same as the branches at (6). And given the match between $\mathsf{I}_{\mathsf{d}}[\mathcal{P}]$ and $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}]$, the bottoms at (3) are the same as the bottom at (5). So there is an S/N pair at (3) iff some branch at (6) is S and the bottom at (5) is N. So $\forall v(\mathcal{Q}(v) \to \mathcal{P})$ and $(\exists v\, \mathcal{Q}(v) \to \mathcal{P})$ are (not) satisfied under the same conditions. By similar reasoning, we are left with the following equivalences to complete our table.

(BI)
$$\begin{array}{lcl}
\forall v(\mathcal{Q}(v) \to \mathcal{P}) & \iff & (\exists v\, \mathcal{Q}(v) \to \mathcal{P}) \\
\exists v(\mathcal{Q}(v) \to \mathcal{P}) & \iff & (\forall v\, \mathcal{Q}(v) \to \mathcal{P})
\end{array}$$

When a universal goes into the antecedent of a conditional, it flips to an existential. And when an existentitial quantifier goes in to the antecedent of a conditional, it flips to a universal. And similarly in the other direction.

Here is an explanation for what is happening: A universal quantifier outside parentheses requires that each inner conditional branch is satisfied; with tips for the consequent $\mathcal{P}$ the same, this requires that either the consequent be S or every antecedent tip be N. But once the quantifier is pushed in, the resultant conditional $\mathcal{A} \to \mathcal{P}$ is satisfied only when the antecedent is N or the consequent is S; so the original requirement that all the antecedent tips be N is matched by the requirement that an *existentential* $\mathcal{A}$ be N. Similarly, an existential quantifier outside parentheses requires that some inner conditional branch is satisfied; with tips for the consequent $\mathcal{P}$ the same, this requires either that the consequent be S or some tip for the antecedent be N. But once the quantifier is pushed in, the resultant conditional $\mathcal{A} \to \mathcal{P}$ is satisfied when the antecedent is N or the consequent is S; and the original requirement that some antecedent tip be N corresponds to the condition that a *universal* $\mathcal{A}$ be N. This case differs from others insofar as the inner conditional branches are S

when the antecedent tips are N. In the standard cases, the branch is S when the tip remains S — and the quantifier goes in as one would expect. The place for caution is when a quantifier comes from or goes into the antecedent of a conditional.[9]

Return to 'everybody loves somebody'. We gave as a translation, $\forall x[Px \to \exists y(Py \wedge Lxy)]$. But $\forall x \exists y[Px \to (Py \wedge Lxy)]$ does as well. To see this, notice that the immediate subformula, $[Px \to \exists y(Py \wedge Lxy)]$ is of the form $[\mathcal{P} \to \exists v \mathcal{Q}(v)]$ where $\mathcal{P}$ has no free instance of the quantified variable y. The quantifier is in the consequent of the conditional, so $[Px \to \exists y(Py \wedge Lxy)]$ is equivalent to $\exists y[Px \to (Py \wedge Lxy)]$. So the larger formula $\forall x[Px \to \exists y(Py \wedge Lxy)]$ is equivalent to $\forall x \exists y[Px \to (Py \wedge Lxy)]$. And similarly in other cases. Officially, there is no reason to prefer one option over the other. Informally, however, there is perhaps less room for confusion when we keep quantifiers relatively close to the expressions they modify. One reason for this is that we continue to associate $\forall$ with $\to$ and $\exists$ with $\wedge$. On this basis, $\forall x[Px \to \exists y(Py \wedge Lxy)]$ is to be preferred. If you have followed this discussion, you are doing well — and should be in a good position to think about the following exercises.

E5.23. Use trees to explain one of the equivalences in table (BC), and one of the equivalences in (BF), for an operator other than $\wedge$. Then use trees to explain the second equivalence in (BI). Be sure to explain how your trees justify the results.

E5.24. Explain why we have not listed quantifier placement equivalences matching $\forall v(\mathcal{P} \leftrightarrow \mathcal{Q}(v))$ with $(\mathcal{P} \leftrightarrow \forall v \mathcal{Q}(v))$. Hint: consider $\forall v(\mathcal{P} \leftrightarrow \mathcal{Q}(v))$ as an abbreviation of $\forall v[(\mathcal{P} \to \mathcal{Q}(v)) \wedge (\mathcal{Q}(v) \to \mathcal{P})]$; from trees, you can see that this is equivalent to $[\forall v(\mathcal{P} \to \mathcal{Q}(v)) \wedge \forall v(\mathcal{Q}(v) \to \mathcal{P})]$. Now, what is the consequence of quantifier placement difficulties for $\to$? Would it work if the quantifier did not flip?

E5.25. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following. (The last generates a famous paradox — can a barber shave himself?)

[9]Thus, for example, we should expect quantifier flipping when pushing into expressions $\forall v(\mathcal{P} \downarrow \mathcal{Q}(v))$ or $\forall v(\mathcal{Q}(v) \downarrow \mathcal{P})$ with a *neither-nor* operator true only when both sides are false. And this is just so: The universal expression is satisfied only when all the inner branches are satisfied; and the inner branches are satisfied just when all the tips are not. And this is like the condition from the existential quantifier in $\exists v \mathcal{Q} \downarrow \mathcal{P}$ or $\mathcal{P} \downarrow \exists v \mathcal{Q}$. And similarly for existentially quantified expressions with this operator.

U: $\{\mathsf{o} \mid \mathsf{o}$ is a person$\}$
b: Bob
B^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a barber$\}$
M^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a man$\}$
S^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m shaves $\mathsf{n}\}$

a. Bob shaves himself.

b. Everyone shaves everyone.

c. Someone shaves everyone.

d. Everyone is shaved by someone.

e. Someone is shaved by everyone.

f. Not everyone shaves themselves.

*g. Any man is shaved by someone.

h. Some man shaves everyone.

i. No man is shaved by all barbers.

*j. Any man who shaves everyone is a barber.

k. If someone shaves all men, then they are a barber.

l. If someone shaves everyone, then they shave themselves.

m. A barber shaves anyone who does not shave themselves.

*n. A barber shaves only people who do not shave themselves.

o. A barber shaves all and only people who do not shave themselves.

E5.26. Given an extended version of $\mathcal{L}_{NT}^{<}$ and the standard interpretation $\mathsf{N1}$ as below, complete the translation for each of the following. Recall that $<$ and $=$ are relation symbols, where S, $\times$ and $+$ are function symbols. As we shall see shortly, it is possible to define E and P in the primitive vocabulary. Also the last sentence states the famous Goldbach conjecture, so far unproved!

U: $\mathbb{N}$

$\emptyset$: zero

S: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$

$+$: $\{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

$\times$: $\{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

$<$: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and m is less than $\mathsf{n}\}$

E^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathbb{N}$ and o is even$\}$

P^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathbb{N}$ and o is prime$\}$

*a. One plus one equals two.

b. Three is greater than two.

c. There is an even prime number.

d. Zero is less than or equal to every number.

e. There is a number less than or equal to every other.

f. For any prime, there is one greater than it.

*g. Any odd (non-even) number is equal to the successor of some even number.

h. Some even number is not equal to the successor of any odd number.

i. A number x is even iff it is equal to two times some y.

j. A number x is odd if it is equal to two time some y plus one.

k. Any odd number is equal to the sum of an odd and an even.

l. Any even number not equal to zero is the sum of one odd with another.

*m. The sum of one odd with another odd is even.

n. No odd number is greater than every prime.

o. Any even number greater than two is equal to the sum of two primes.

E5.27. Produce a good quantificational translation for each of the following. In this case you should provide an interpretation function for the sentences. Let U be the set of people, and, assuming that each has a unique best friend, implement a *best friend of* function.

a. Bob's best friend likes all New Yorkers.

b. Some New Yorker likes all Californians.

c. No Californian likes all New Yorkers.

d. Any Californian likes some New Yorker.

e. Californians who like themselves, like at least some people who do not.

f. New Yorkers who do not like themselves, do not like anybody.

g. Nobody likes someone who does not like them.

h. There is someone who dislikes every new Yorker, and is liked by every Californian.

i. Anyone who likes themselves and dislikes every New Yorker, is liked by every Californian.

j. Everybody who likes Bob's best friend likes some New Yorker who does not like Bob.

5.3.4 Equality

We complete our discussion of translation by turning to some important applications for equality. Adopt an interpretation function with U the set of people and,

b: Bob

c: Bob

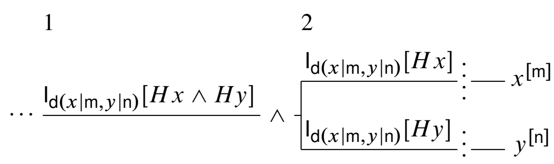
f^1: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$, where n is the father of $\mathsf{m}\}$

H^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is a happy person$\}$

(Maybe Bob's friends call him "Cronk.") The simplest applications for $=$ assert the identity of individuals. Thus, for example, $b = c$ is satisfied insofar as $\langle \mathsf{I_d}[b], \mathsf{I_d}[c]\rangle \in \mathsf{I}[=]$. Similarly, $\exists x(b = f^1x)$ is satisfied just in case Bob is someone's father. And, on the standard interpretation of $\mathscr{L}_{NT}^{<}$, $\exists x[(x + x) = (x \times x)]$ is satisfied insofar as, say, $\langle \mathsf{I}_{\mathsf{d}(x|2)}[x + x], \mathsf{I}_{\mathsf{d}(x|2)}[x \times x]\rangle \in \mathsf{I}[=]$ — that is, $\langle 4, 4\rangle \in \mathsf{I}[=]$. If this last case is not clear, think about it on a tree.

We get to an interesting class of cases when we turn to *quantity* expressions. Thus, for example, we can easily say 'at least one person is happy', $\exists xHx$. But notice that neither $\exists xHx \wedge \exists yHy$ nor $\exists x\exists y(Hx \wedge Hy)$ work for 'at least two people are happy'. For the first, it should be clear that each conjunct is satisfied, so that the conjunction is satisfied, so long as there is at least one happy person. And similarly for the second. To see this in a simple case, suppose Bob, Sue and Jim are the only people in U. Then the existentials for $\exists x\exists y(Hx \wedge Hy)$ result in nine branches of the following sort,

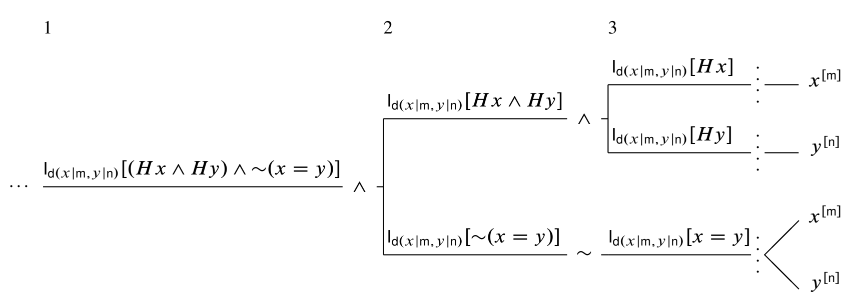
(BJ)



for some individuals m and n. Just one of these branches has to be satisfied in order for the main sentence to be satisfied and true. Clearly none of the tips are satisfied if none of Bob, Sue or Jim is happy; then the branches are N and $\exists x\exists y(Hx \wedge Hy)$ is N as well. But suppose just one of them, say Sue, is happy. Then on the branch for $\mathsf{d}_{(x|\text{Sue},y|\text{Sue})}$ *both* Hx and Hy are satisfied! Thus the conjunction is satisfied, and the existential is satisfied as well. So $\exists x\exists y(Hx \wedge Hy)$ does not require that at least two people are happy. The problem, again, is that the same person might satisfy both conjuncts at once.

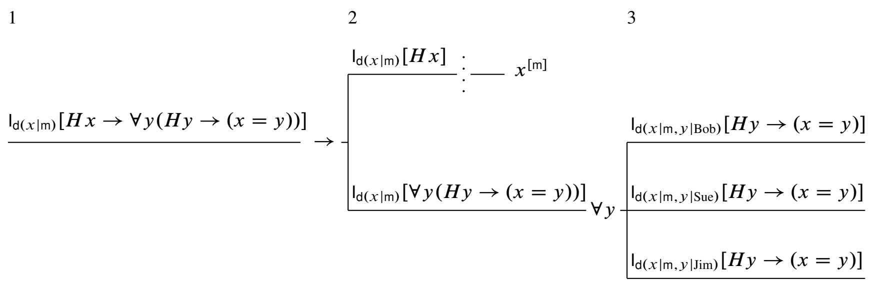
But this case points the way to a good translation for 'at least two people are happy'. We get the right result with, $\exists x\exists y[(Hx \wedge Hy) \wedge {\sim}(x = y)]$. Now, in our simple example, the existentials result in nine branches as follows,

(BK)



The sentence is satisfied and true if at least one branch is satisfied. Now in the case where just Sue is happy, on the branch with $\mathsf{d}_{(x|\text{Sue},y|\text{Sue})}$ both Hx and Hy are satisfied as before. But this branch has $x = y$ satisfied; so $\sim(x = y)$ is not satisfied, and the branch as a whole fails. But suppose both Bob and Sue are happy. Then on the branch with $\mathsf{d}_{(x|\text{Bob},y|\text{Sue})}$ both Hx and Hy are satisfied; but this time, $x = y$ is not satisfied; so $\sim(x = y)$ is satisfied, and the branch is satisfied, so that the whole sentence, $\exists x \exists y[(Hx \wedge Hy) \wedge \sim(x = y)]$ is satisfied and true. That is, the sentence is satisfied and true just when the happy people assigned to x and y are distinct — just when there are at least two happy people. On this pattern, you should be able to see how to say there are at least three happy people, and so forth.

Now suppose we want to say, 'at most one person is happy'. We have, of course, learned a couple of ways to say nobody is happy, $\forall x{\sim}Hx$ and $\sim\exists xHx$. But for 'at most one' we need something like, $\forall x[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$. For this, in our simplified case, the universal quantifier yields three branches of the sort, $\mathsf{I}_{\mathsf{d}(x|\mathsf{m})}[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$. The beginning of the branch is as follows,

(BL) 

The universal $\forall x[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$ is satisfied and true if and only if all the conditional branches at (1) are satisfied. And the branchs at (1) are satisfied so long as there is no S/N pair at (2). This is of course so if nobody is happy so that the top at (2) is never satisfied. But suppose m is a happy person, say, Sue and the top at (2) is satisfied. The bottom comes out S so long as Sue is the only happy person, so that any happy y is identical to her. In this case, again, we do not get an S/N pair. But suppose Jim, say, is also happy; then the very bottom branch at (3) fails; so the universal at (2) is N; so the conditional at (1) is N; and the entire sentence is N. Suppose x is assigned to a happy person; in effect, $\forall y(Hy \rightarrow (x = y))$ limits the range of happy things, telling us that *anything happy is it*. We get 'at most two people are happy' with $\forall x \forall y[(Hx \wedge Hy) \rightarrow \forall z(Hz \rightarrow (x = z \vee y = z))]$ — if some things are happy, then anything that is happy is one of them. And similarly in other cases.

To say 'exactly one person is happy, it is enough to say at least one person is happy, and at most one person is happy. Thus, using what we have already done, $\exists x Hx \wedge \forall x[Hx \rightarrow \forall y(Hy \rightarrow (x = y))]$ does the job. But we can use the "limiting" strategy with the universal quantifier more efficiently. Thus, for example, if we want to say, 'Bob is the only happy person' we might try $Hb \wedge \forall y[Hy \rightarrow (b = y)]$ — Bob is happy, and every happy person *is* Bob. Similarly, for 'exactly one person is happy', $\exists x[Hx \wedge \forall y(Hy \rightarrow (x = y))]$ is good. We say that there is a happy person, and that all the happy people are identical to it. For 'exactly two people are happy', $\exists x \exists y[((Hx \wedge Hy) \wedge \sim(x = y)) \wedge \forall z(Hz \rightarrow [(x = z) \vee (y = z)])]$ does the job — there are at least two happy people, and anything that is a happy person is identical to one of them.

Phrases of the sort "the such-and-such" are *definite descriptions*. Perhaps it is natural to think "the such-and-such is so-and-so" *fails* when there is more than one such-and-such. Similarly, phrases of the sort "the such-and-such is so-and-so" seem to fail when nothing is such-and-such. Thus, for example, neither 'The desk at CSUSB has graffiti on it' nor 'the present king of France is bald' seem to be true. The first because the description fails to pick out just one object, and the second because the description does not pick out any object. Of course, if a description does pick out just one object, then the predicate must apply. So, for example, as I write, 'The president of the USA is a woman' is not true. There is exactly one object which is the president of the USA, but it is not a woman. And 'the president of the USA is a man' is true. In this case, exactly one object is picked out by the description, and the predicate does apply. Thus, in "On Denoting," Bertrand Russell famously proposes that a statement of the sort 'the $\mathcal{P}$ is $\mathcal{Q}$' is true just in case there is exactly one $\mathcal{P}$ and it is $\mathcal{Q}$. On Russell's account, then, where $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ have variable x free, and $\mathcal{P}(v)$ is like $\mathcal{P}(x)$ but with free instances of x replaced by a new variable v, $\exists x[(\mathcal{P}(x) \wedge \forall v(\mathcal{P}(v) \rightarrow x = v)) \wedge \mathcal{Q}(x)]$ is good — there is a $\mathcal{P}$, it is the only $\mathcal{P}$, and it is $\mathcal{Q}$. Thus, for example, with the natural interpretation function, $\exists x[(Px \wedge \forall y(Py \rightarrow x = y)) \wedge Wx]$ translates 'the president is a woman'. In a course on philosophy of language, one might spend a great deal of time discussing definite descriptions. But in ordinary cases we will simply assume Russell's account for translating expressions of the sort, 'the $\mathcal{P}$ is $\mathcal{Q}$'.

Finally, notice that equality can play a role in *exception* clauses. This is particularly important when making general comparisons. Thus, for example, if we want to say that zero is smaller than every other integer, with the standard interpretation N1 of $\mathcal{L}_{NT}^{<}$, $\forall x(\emptyset < x)$ is a mistake. This formula is satisfied only if zero is less than zero! What we want is rather, $\forall x[\sim(x = \emptyset) \rightarrow (\emptyset < x)]$. Similarly, if we want to say that there is a person taller than every other, we would not use $\exists x \forall y Txy$ where

Txy when x is taller than y. This would require that the tallest person be taller than herself! What we want is rather, $\exists x \forall y[\sim(x = y) \rightarrow Txy]$.

Observe that relations of this sort may play a role in definite descriptions. Thus it seems natural to talk about *the* smallest integer, or *the* tallest person. We might therefore additionally assert uniqueness with something like, $\exists x[x$ is taller than every other $\wedge\ \forall z($z is taller than every other $\rightarrow x = z)]$.[10] However, we will not usually add the second clause, insofar as uniqueness follows automatically in these cases from the initial claim, $\exists x \forall y[\sim(x = y) \rightarrow Txy]$ together with the premise that *taller than* (*less than*) is asymmetric, that $\forall x \forall y(Txy \rightarrow \sim Tyx)$.[11] By itself, $\exists x \forall y[\sim(x = y) \rightarrow Txy]$ does not require uniqueness — it says only that there is a tallest object. When a relation is asymmetric, however, there cannot be multiple things with the relation to everything else. Thus, in these cases, for 'The tallest person is happy' it will be sufficient conjoin 'a tallest person is happy' with asymmetry, $\exists x[\forall y(\sim(x = y) \rightarrow Txy) \wedge Hx] \wedge \forall x \forall y(Txy \rightarrow \sim Tyx)$. Taken together, these imply all the elements of Russell's account.

E5.28. Given the following partial interpretation function for $\mathcal{L}_q$, complete the translation for each of the following.

U: $\{\mathsf{o} \mid \mathsf{o}$ is a a snake in my yard$\}$

a: Aaalph

G^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is in the grass$\}$

D^1: $\{\mathsf{o} \mid \mathsf{o} \in \mathsf{U}$ and o is deadly$\}$

B^2: $\{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m}, \mathsf{n} \in \mathsf{U}$ and m is bigger than $\mathsf{n}\}$

a. There is at least one snake in the grass.

b. There are at least two snakes in the grass.

*c. There are at least three snakes in the grass.

d. There are no snakes in the grass.

e. There is at most one snake in the grass.

[10] $\exists x[\forall y(\sim(x = y) \rightarrow Txy) \wedge \forall z(\forall y(\sim(z = y) \rightarrow Tzy) \rightarrow x = z)]$.

[11] If m is taller than everything other than itself, n is taller than everything other than itself, but $\mathsf{m} \neq \mathsf{n}$, then m is taller than n and n is taller than m. But this is impossible if the relation is asymmetric. So only one object can be taller than all the others.

f. There are at most two snakes in the grass.

g. There are at most three snakes in the grass.

h. There is exactly one snake in the grass.

i. There are exactly two snakes in the grass.

j. There are exactly three snakes in the grass.

*k. The snake in the grass is deadly.

l. Aaalph is the biggest snake.

*m. Aaalph is bigger than any other snake in the grass.

n. The biggest snake in the grass is deadly.

o. The smallest snake in the grass is deadly.

E5.29. Given $\mathscr{L}_{NT}^{<}$ and a function for the standard interpretation as below, complete the translation for each of the following. Hint: Once you know how to say a number is odd or even, answers to some exercises will mirror ones from E5.26.

U: $\mathbb{N}$
$\emptyset$: zero
S: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$
$+$: $\{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$
$\times$: $\{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$
$<$: $\{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and m is less than $\mathsf{n}\}$

a. Any number is equal to itself (identity is *reflexive*).

b. If a number a is equal to a number b, then b is equal to a (identity is *symmetric*).

c. If a number a is equal to a number b and b is equal to c, then a is equal to c (identity is *transitive*).

d. No number is less than itself (less than is *irreflexive*).

*e. If a number a is less than a number b, then b is not less then a (less than is *asymmetric*).

f. If a number a is less than a number b and b is less than c, then a is less than c (less than is *transitive*).

g. There is no largest number.

*h. Four is even (a number such that two times something is equal to it).

i. Three is odd (such that two times something plus one is equal to it).

*j. Any odd number is the sum of an odd and an even.

k. Any even number other than zero is the sum of one odd with another.

l. The sum of one odd with another odd is even.

m. There is no largest even number.

*n. Three is prime (a number divided by no number other than one and itself — though you will have to put this in terms of multipliers).

o. Every prime except two is odd.

E5.30. For each of the following arguments: (i) Produce a good translation, including interpretation function and translations for the premises and conclusion. Then (ii) for each argument that is not quantificationally valid, produce an interpretation (trees optional) to show that the argument is not quantificationally valid.

a. Only citizens can vote
Hannah is a citizen
———
Hannah can vote

b. All citizens can vote
If someone is a citizen, then their father is a citizen
Hannah is a citizen
———
Hannah's father can vote

*c. Bob is taller than every other man
———
Only Bob is taller than every other man

d. Bob is taller than every other man
The *taller than* relation is asymmetric
———
Only Bob is taller than every other man

e. Some happy animals are dogs
At most one happy dog is chasing a cat
———
Some happy dog is chasing a cat

E5.31. For each of the arguments in E530 that you have not shown is invalid, produce a derivation to show that it is valid in *AD*.

E5.32. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Quantifier switching

b. Quantifier placement

c. Quantity expressions and definite descriptions

Chapter 6

Natural Deduction

Natural deductions systems are so-called because their rules formalize patterns of reasoning that occur in relatively ordinary "natural" contexts. Thus, initially at least, the rules of natural deduction systems are easier to motivate than the axioms and rules of axiomatic systems. By itself, this is sufficient to give natural deduction a special interest. As we shall see, natural deduction is also susceptible to proof *strategies* in a way that (primitive) axiomatic systems are not. If you have had another course in formal logic, you have probably been exposed to natural deduction. So, again, it may seem important to bring what we have done into contact with what you have encountered in other contexts. After some general remarks about natural deduction, we turn to the sentential and quantificational components of our system *ND*, and finally to an expanded system, *ND*+.

6.1 General

I begin this section with a few general remarks about derivation systems and derivation rules. We will then turn to some background notions for the particular rules of our official natural derivation systems.[1]

6.1.1 Derivations as Games

In their essential nature, derivations are defined in terms of form. Both axiomatic and natural derivations can be seen as a kind of game — with the aim of getting from a starting point to a goal by rules. Taken as games, there is no immediate or obvious

[1]Parts of this section are reminiscent of 3.1 and, especially if you skipped over that section, you may want to look over it now as additional background.

connection between derivations and semantic validity or truth. This point may have been particularly vivid with respect to axiomatic systems. In the case of natural derivations, the systems are driven by *rules* rather than axioms, and the rules may "make sense" in a way that axioms do not. Still, we can introduce natural derivations purely in their nature as games. Thus, for example, consider a system $N1$ with the following rules.

$$N1 \qquad \text{R1}\ \frac{\mathcal{P} \rightarrow \mathcal{Q},\ \mathcal{P}}{\mathcal{Q}} \qquad \text{R2}\ \frac{\mathcal{P} \vee \mathcal{Q}}{\mathcal{Q}} \qquad \text{R3}\ \frac{\mathcal{P} \wedge \mathcal{Q}}{\mathcal{P}} \qquad \text{R4}\ \frac{\mathcal{P}}{\mathcal{P} \vee \mathcal{Q}}$$

In this system, R1: given formulas of the form $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$, one may move to $\mathcal{Q}$; R2: given a formula of the form $\mathcal{P} \vee \mathcal{Q}$, one may move to $\mathcal{Q}$; R3: given a formula of the form $\mathcal{P} \wedge \mathcal{Q}$, one may move to $\mathcal{P}$; and R4: given a formula $\mathcal{P}$ one may move to $\mathcal{P} \vee \mathcal{Q}$ for any $\mathcal{Q}$. For now, at least, the game is played as follows: One begins with some starting formulas and a goal. The starting formulas are like "cards" in your hand. One then applies the rules to obtain more formulas, to which the rules may be applied again and again. You win if you eventually obtain the goal formula. Each application of a rule is *independent* of the ones before — so all that matters for a given move is whether formulas are of the requisite forms; it does not matter what was $\mathcal{P}$ or what was $\mathcal{Q}$ in a previous application of the rules.

Let us consider some examples. At this stage, do not worry about strategy, about why we do what we do, as much as about how the rules work and the way the game is played. A game always begins with starting premises at the top, and goal on the bottom.

(A)

1.	$A \rightarrow (B \wedge C)$	P(remise)
2.	A	P(remise)
	$B \vee D$	(goal)

The formulas on lines (1) and (2) are of the form $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$, where $\mathcal{P}$ maps to A and $\mathcal{Q}$ to $(B \wedge C)$; so we are in a position to apply rule R1 to get the $\mathcal{Q}$.

1.	$A \rightarrow (B \wedge C)$	P(remise)
2.	A	P(remise)
3.	$B \wedge C$	1,2 R1
	$B \vee D$	(goal)

The justification for our move — the way the rules apply — is listed on the right; in this case, we use the formulas on lines (1) and (2) according to rule R1 to get $B \wedge C$;

so that is indicated by the notation. Now, $B \wedge C$ is of the form $\mathcal{P} \wedge \mathcal{Q}$. So we can apply R3 to it in order to obtain the $\mathcal{P}$, namely B.

1.	$A \rightarrow (B \wedge C)$	P(remise)
2.	A	P(remise)
3.	$B \wedge C$	1,2 R1
4.	B	3 R3
	$B \vee D$	(goal)

Notice that one application of a rule is independent of another. It does not matter what formula was $\mathcal{P}$ or $\mathcal{Q}$ in a previous move, for evaluation of this one. Finally, where $\mathcal{P}$ is B, $B \vee D$ is of the form $\mathcal{P} \vee \mathcal{Q}$. So we can apply R4 to get the final result.

1.	$A \rightarrow (B \wedge C)$	P(remise)
2.	A	P(remise)
3.	$B \wedge C$	1,2 R1
4.	B	3 R3
5.	$B \vee D$	4 R4 Win!

Notice that R4 leaves the $\mathcal{Q}$ unrestricted: Given some $\mathcal{P}$, we can move to $\mathcal{P} \vee \mathcal{Q}$ for *any* $\mathcal{Q}$. Since we reached the goal from the starting sentences, we win! In this simple derivation system, any line of a successful derivation is a premise, or justified from lines before by the rules.

Here are a couple more examples, this time of completed derivations.

(B)

1.	$A \wedge C$	P
2.	$(A \vee B) \rightarrow D$	P
3.	A	1 R3
4.	$A \vee B$	3 R4
5.	D	2,4 R1
6.	$D \vee (R \rightarrow S)$	5 R4 Win!

$A \wedge C$ is of the form $\mathcal{P} \wedge \mathcal{Q}$. So we can apply R3 to obtain the $\mathcal{P}$, in this case A. Then where $\mathcal{P}$ is A, we use R4 to add on a B to get $A \vee B$. $(A \vee B) \rightarrow D$ and $A \vee B$ are of the form $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$; so we apply R1 to get the $\mathcal{Q}$, that is D. Finally, where D is $\mathcal{P}$, $D \vee (R \rightarrow S)$ is of the form $\mathcal{P} \vee \mathcal{Q}$; so we apply R4 to get the final result. Notice again that the $\mathcal{Q}$ may be any formula whatsoever.

Here is another example.

(C)

1.	$(A \wedge B) \wedge D$	P
2.	$(A \wedge B) \rightarrow C$	P
3.	$A \rightarrow (C \rightarrow (B \wedge D))$	P
4.	$A \wedge B$	1 R3
5.	C	2,4 R1
6.	A	4 R3
7.	$C \rightarrow (B \wedge D)$	3,6 R1
8.	$B \wedge D$	7,5 R1
9.	B	8 R3 Win!

You should be able to follow the steps. In this case, we use $A \wedge B$ on line (4) twice; once as part of an application of R1 to get C, and again in an application of R3 to get the A. Once you have a formula in your "hand" you can use it as many times and whatever way the rules will allow. Also, the order in which we worked might have been different. Thus, for example, we might have obtained A on line (5) and then C after. You win if you get to the goal by the rules; how you get there is up to you. Finally, it is tempting to think we could get B from, say, $A \wedge B$ on line (4). We will able to do this in our official system. But the rules we have so far do not let us do so. R3 lets us move just to the left conjunct of a formula of the form $\mathcal{P} \wedge \mathcal{Q}$.

When there is a way to get from the premises of some argument to its conclusion by the rules of derivation system *N*, the premises *prove* the conclusion in system *N*. In this case, where Γ (Gamma) is the set of premises, and $\mathcal{P}$ the conclusion we write $\Gamma \vdash_N \mathcal{P}$. If $\Gamma \vdash_N \mathcal{P}$ the argument is *valid* in derivation system *N*. Notice the distinction between this "single turnstile" $\vdash$ and the double turnstile $\vDash$ associated with semantic validity. As usual, if $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ are the members of Γ, we sometimes write $\mathcal{Q}_1 \ldots \mathcal{Q}_n \vdash_N \mathcal{P}$ in place of $\Gamma \vdash_N P$. If Γ has no members then, listing all the members of Γ individually, we simply write $\vdash_N \mathcal{P}$. In this case, $\mathcal{P}$ is a *theorem* of derivation system *N*.

One can imagine setting up many different rule sets, and so many different games of this kind. In the end, we want our game to serve a specific purpose. That is, we want to use the game in the identification of valid arguments. In order for our games to be an indicator of validity, we would like it to be the case that $\Gamma \vdash_N \mathcal{P}$ iff $\Gamma \vDash \mathcal{P}$, that Γ *proves* $\mathcal{P}$ iff Γ *entails* $\mathcal{P}$. In Part III we will show that our official derivation games have this property.

For now, we can at least see how this might be: Roughly, we impose the following condition on rules: we require of our rules that *the inputs always semantically entail the outputs*. Then if some premises are true, and we make a move to a formula, the formula we move to must be true; and if the formulas in our "hand" are all true, and we add some formula by another move, the formula we add must be true; and so

forth for each formula we add until we get to the goal, which will have to be true as well. So if the premises are true, the goal must be true as well. We will have much more to say about this later!

For now, notice that our rules R1, R3 and R4 each meet the proposed requirement on rules, but R2 does not.

(D)

		R1			R2		R3		R4	
$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \to \mathscr{Q}$	$\mathscr{P}$ /	$\mathscr{Q}$	$\mathscr{P} \vee \mathscr{Q}$ /	$\mathscr{Q}$	$\mathscr{P} \wedge \mathscr{Q}$ /	$\mathscr{P}$	$\mathscr{P}$ /	$\mathscr{P} \vee \mathscr{Q}$
T	T	***T***	***T***	***T***	***T***	***T***	***T***	***T***	***T***	***T***
T	F	***F***	***T***	***F***	***T***	***F***	***F***	***T***	***T***	***T***
F	T	***T***	***F***	***T***	***T***	***T***	***F***	***F***	***F***	***T***
F	F	***T***	***F***	***F***	***F***	***F***	***F***	***F***	***F***	***F***

R1, R3 and R4 have no row where the input(s) are T and the output is F. But for R2, the second row has input T and output F. So R2 does not meet our condition. This does not mean that one cannot construct a *game* with R2 as a part. Rather, the point is that R2 will not help us accomplish what we want to accomplish with our games. As we demonstrate in Part III, so long as rules meet the condition, a win in the game always corresponds to an argument that is semantically valid. Thus, for example, derivation (C), in which R2 does not appear, corresponds to the result that $(A \wedge B) \wedge D, (A \wedge B) \to C, A \to (C \to (B \wedge D)) \vDash_s B$.

(E)

A	B	C	D	$(A \wedge B)$	$\wedge\, D$	$(A \wedge B)$	$\to C$	$A \to$	$(C \to$	$(B \wedge D))$	/ B
T	T	T	T	T	***T***	T	***T***	***T***	T	T	***T***
T	T	T	F	T	***F***	T	***T***	***F***	F	F	***T***
T	T	F	T	T	***T***	T	***F***	***T***	T	T	***T***
T	T	F	F	T	***F***	T	***F***	***T***	T	F	***T***
T	F	T	T	F	***F***	F	***T***	***F***	F	F	***F***
T	F	T	F	F	***F***	F	***T***	***F***	F	F	***F***
T	F	F	T	F	***F***	F	***T***	***T***	T	F	***F***
T	F	F	F	F	***F***	F	***T***	***T***	T	F	***F***
F	T	T	T	F	***F***	F	***T***	***T***	T	T	***T***
F	T	T	F	F	***F***	F	***T***	***T***	F	F	***T***
F	T	F	T	F	***F***	F	***T***	***T***	T	T	***T***
F	T	F	F	F	***F***	F	***T***	***T***	T	F	***T***
F	F	T	T	F	***F***	F	***T***	***T***	F	F	***F***
F	F	T	F	F	***F***	F	***T***	***T***	F	F	***F***
F	F	F	T	F	***F***	F	***T***	***T***	T	F	***F***
F	F	F	F	F	***F***	F	***T***	***T***	T	F	***F***

There is no row where the premises are T and the conclusion is F. As the number of rows goes up, we may decide that the games are dramatically easier to complete than the tables. And derivations are particularly important in the quantificational case, where we have not yet been able to demonstrate semantic validity at all.

E6.1. Show that each of the following is valid in $N1$. Complete (a) - (d) using just rules R1, R3 and R4. You will need an application of R2 for (e).

*a. $(A \wedge B) \wedge C \vdash_{N1} A$

b. $(A \wedge B) \wedge C, A \rightarrow (B \wedge C) \vdash_{N1} B$

c. $(A \wedge B) \rightarrow (B \wedge A), A \wedge B \vdash_{N1} B \vee A$

d. $R, [R \vee (S \vee T)] \rightarrow S \vdash_{N1} S \vee T$

e. $A \vdash_{N1} A \rightarrow C$

*E6.2. (i) For each of the arguments in E6.1, use a truth table to decide if the argument is sententially valid. (ii) To what do you attribute the fact that a win in $N1$ is not a sure indicator of semantic validity?

6.1.2 Auxiliary Assumptions

So far, our derivations have had the following form,

$$
\text{(F)}\qquad
\begin{array}{r|ll}
\text{a.} & \mathcal{A} & \text{P(remise)} \\
 & \vdots & \\
\text{b.} & \mathcal{B} & \text{P(remise)} \\
\cline{2-2}
 & \vdots & \\
\text{c.} & \mathcal{G} & \text{(goal)}
\end{array}
$$

We have some premise(s) at the top, and a conclusion at the bottom. The premises are against a line which indicates the range or *scope* over which the premises apply. In each case, the line extends from the premises to the conclusion, indicating that the conclusion is derived from them. It is always our aim to derive the conclusion under the scope of the premises alone. But our official derivation system will allow appeal to certain *auxiliary* assumptions in addition to premises. Any such assumption comes with a scope line of its own — indicating the range over which *it* applies. Thus, for example, derivations might be structured as follows.

(G)

a.	$\mathcal{A}$	P(remise)
b.	$\mathcal{B}$	P(remise)
c.	$\mathcal{C}$	A(ssumption)
d.		
e.	$\mathcal{G}$	(goal)

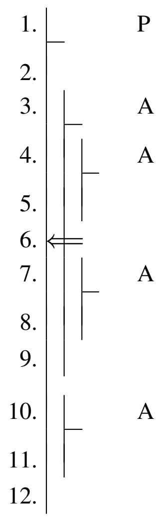
(H)

a.	$\mathcal{A}$	P(remise)
b.	$\mathcal{B}$	P(remise)
c.	$\mathcal{C}$	A(ssumption)
d.	$\mathcal{D}$	A(ssumption)
e.		
f.		
g.	$\mathcal{G}$	(goal)

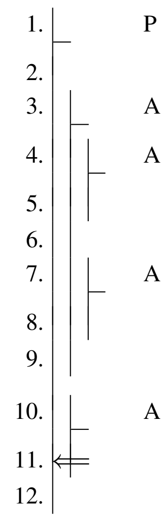
In each, there are premises $\mathcal{A}$ through $\mathcal{B}$ at the top and goal $\mathcal{G}$ at the bottom. As indicated by the main leftmost scope line, the premises apply throughout the derivations, and the goal is derived under them. In case (G), there is an additional assumption at (c). As indicated by its scope line, that assumption applies from (c) - (d). In (H), there are a pair of additional assumptions. As indicated by the associated scope lines, the first applies over (c) - (f), and the second over (d) - (e). We will say that an auxiliary assumption, together with the formulas that fall under its scope, is a *subderivation*. Thus (G) has a subderivation on from (c) - (d). (H) has a pair of subderivations, one on (c) - (f), and another on (d) - (e). A derivation or subderivation may *include* various other subderivations. Any subderivation begins with an auxiliary assumption. In general we *cite* a subderivation by listing the line number on which it begins, then a dash, and the line number on which its scope line ends.

In contexts without auxiliary assumptions, we have been able freely to appeal to any formula already in our "hand." Where there are auxiliary assumptions, however, we may appeal only to *accessible* subderivations and formulas. A formula is *accessible* at a given stage when it is obtained under assumptions all of which continue to apply. In practice, what this means is that for justification of a formula at line number i we can appeal only to formulas which appear immediately against scope lines extending as far as i. Thus, for example, with the scope structure as in (I) below, in the justification of line (6),

(I)



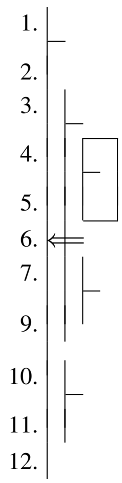
(J)



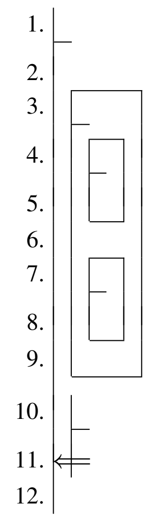
we could appeal only to formulas at (1), (2) and (3), for these are the only ones immediately against scope lines extending as far as (6). To see this, notice that scope lines extending as far as (6), are ones cut by the arrow at (6). Formulas at (4) and (5) are not against a line extending that far. Similarly, as indicated by the arrow in (J), for the justification of (11), we could appeal only to formulas at (1), (2), and (10). Formulas at other line numbers are not immediately against scope lines extending as far as (11). The accessible formulas are ones derived under assumptions all of which continue to apply.

It may be helpful to think of a completed subderivation as a sort of "box." So long as you are under the scope of an assumption, the box is open and you can "see" the formulas under its scope. However, once you exit from an assumption, the box is closed, and the formulas inside are no longer available.

(I′)



(J′)



Thus, again, in (I′) the formulas at (4) - (5) are locked away so that the only accessible lines are (1) - (3). Similarly, at line (11) of (J′) all of (3) - (9) is unavailable.

Our aim is always to obtain the goal against the leftmost scope line — under the scope of the premises alone — and if the only formulas accessible for its justification are also against the leftmost scope line, it may appear mysterious why we would ever introduce auxiliary assumptions and subderivations at all. What is the point of auxiliary assumptions, if formulas under their scope are inaccessible for justification for the formula we want? The answer is that, though the formulas inside a box are unavailable *the box* may still be useful. Certain of our rules will appeal to entire subderivations (to the boxes), rather than to the formulas in them. A subderivation is *accessible* at a given stage when *it* is obtained under assumptions all of which continue to apply. In practice, what this means is that for a formula at line i, we can appeal to a box (to a subderivation) only if *it* (its scope line) is against a line which extends down to i.

Thus at line (6) of (I′), we would not be able to appeal to the formulas on lines (4) and (5) — they are inside the closed box. However, we *would* be able to appeal to the *box* on lines (4) - (5), for *it* is against a scope line cut by the arrow. Similarly, at line (11) of (J′) we are not able to appeal to formulas on any of the lines (3) - (9), for they are inside the closed boxes. Similarly, we cannot appeal to the *boxes* on (4) - (5) or (7) - (8) for they are locked inside the larger box. However, we can appeal to the larger subderivation on (3) - (9) insofar as it is against a line cut by the arrow. Observe that one can appeal to a box only after it is closed – so, for example, at (11) of (J′) there is not (yet) a closed box at (10) - (11) and so no available subderivation to which one may appeal.

Putting this together, at (12) we can appeal to the subderivations at (3) - (9) and (10) - (11); the ones at (4) - (5) and (7) - (8) remain inaccessible. The justification for line (12) might therefore appeal to the formulas on lines (1) and (2) or to the subderivations on lines (3) - (9) and (10) - (11). Again line (12) does not have access to the *formulas* inside the subderivations from lines (3) - (9) and (10) - 11). So the subderivations are accessible even where the formulas inside them are not, and there may be a point to the subderivations even where the formulas *inside* the subderivation are inaccessible.

Definitions for Auxiliary Assumptions

SD An auxiliary assumption, together with the formulas that fall under its scope, is a *subderivation*.

FA A formula is *accessible* at a given stage when it is obtained under assumptions all of which continue to apply.

SA A subderivation is *accessible* at a given stage when it (as a whole) is obtained under assumptions all of which continue to apply.

In practice, what this means is that for justification of a formula at line i we can appeal to another formula only if it is immediately against a scope line extending as far as i.

And in practice, for justification of a formula at line i, we can appeal to a subderivation only if its whole *scope line* is itself immediately against a scope line extending as far as i.

All this will become more concrete as we turn now to the rules of our official system *ND*. We can reinforce the point about accessibility of *formulas* by introducing the first, and simplest, rule of our official system. If a formula $\mathcal{P}$ appears on an accessible line a of a derivation, we may repeat it by the rule *reiteration*, with justification a R.

R

a.	$\mathcal{P}$	
	$\mathcal{P}$	a R

It should be obvious why reiteration satisfies our basic condition on rules. If $\mathcal{P}$ is true, *of course* $\mathcal{P}$ is true. So this rule could never lead from a formula that is true, to one that is not. Observe, though, that the line a must be *accessible*. If in (I) the assumption at line (3) were a formula $\mathcal{P}$, then we could conclude $\mathcal{P}$ with justification 3 R at lines (5), (6), (8) or (9). We could not obtain $\mathcal{P}$ with the same justification at (11) or (12) without violating the rule, because (3) is not accessible for justification of (11) or (12). You should be clear about why this is so.

*E6.3. Consider a derivation with the following structure.

1. P
2. A
3.
4. A
5. A
6.
7.
8.

For each of the lines (3), (6), (7) and (8) which lines are accessible? which subderivations (if any) are accessible? That is, complete the following table.

	accessible lines	accessible subderivations
line 3		
line 6		
line 7		
line 8		

*E6.4. Suppose in a derivation with structure as in E6.3 we have obtained a formula $\mathcal{A}$ on line (3). (i) On what lines would we be allowed to conclude $\mathcal{A}$ by 3 R? Suppose there is a formula $\mathcal{B}$ on line (4). (ii) On what lines would we be allowed to conclude $\mathcal{B}$ by 4 R? Hint: this is just a question about accessibility, asking where it is possible to use lines (3) and (4).

6.2 Sentential

Our system *N*1 set up the basic idea of derivations as games. We begin presentation of our official natural deduction system *ND* with rules whose application is just to sentential forms — to forms involving $\sim$, and $\rightarrow$ (and so to $\wedge$, $\vee$, and $\leftrightarrow$). Though the only operators in the forms are sentential, the forms may apply to expressions in either a sentential language like $\mathcal{L}_s$, or a quantificational one like $\mathcal{L}_q$. For the most part, though, we simply focus on $\mathcal{L}_s$. In a derivation, each formula is either a premise, an auxiliary assumption, or is justified by the rules. As we will see, auxiliary assumptions are always introduced in conjunction with an *exit strategy*. In addition to reiteration, the sentential part of *ND* includes two rules for each of the five sentential operators — for a total of eleven rules. For each of the operators, there is an ‘I’

or *introduction* rule, and an 'E' or *exploitation* rule.[2] As we will see, this division helps structure the way we approach derivations: To generate a formula with main operator $\star$, you will typically use the corresponding introduction rule. To make use of a formula with main operator $\star$, you will typically employ the exploitation rule for that operator.

6.2.1 $\rightarrow$ and $\wedge$

Let us start with the I- and E-rules for $\rightarrow$ and $\wedge$. We have already seen the exploitation rule for $\rightarrow$. It is R1 of system $N1$. If formulas $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ and appear on accessible lines a and b of a derivation, we may conclude $\mathcal{Q}$ with justification *a,b* $\rightarrow$E.

$$\rightarrow\text{E} \qquad \begin{array}{r|ll} \text{a.} & \mathcal{P} \rightarrow \mathcal{Q} & \\ \text{b.} & \mathcal{P} & \\ & & \\ & \mathcal{Q} & \text{a,b} \rightarrow\text{E} \end{array}$$

Intuitively, if it is true that *if* $\mathcal{P}$ *then* $\mathcal{Q}$, and it is true that $\mathcal{P}$, then $\mathcal{Q}$ must be true as well. And, on table (D) we saw that if both $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ are true, then $\mathcal{Q}$ is true. Notice that we *do not* somehow get the $\mathcal{P}$ from $\mathcal{P} \rightarrow \mathcal{Q}$. Rather, we exploit $\mathcal{P} \rightarrow \mathcal{Q}$ when, given that $\mathcal{P}$ also is true, we use $\mathcal{P}$ together with $\mathcal{P} \rightarrow \mathcal{Q}$ to conclude $\mathcal{Q}$. So this rule requires two input "cards." The $\mathcal{P} \rightarrow \mathcal{Q}$ card sits idle without a $\mathcal{P}$ to activate it. The order in which $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ appear does not matter so long as they are both accessible. However, you should cite them in the standard order — line for the conditional first, then the antecedent. As in the axiomatic system from chapter 3, this rule is sometimes called *modus ponens*.

Here is an example. We show, L, $L \rightarrow (A \wedge K)$, $(A \wedge K) \rightarrow (L \rightarrow P) \vdash_{ND} P$.

$$\text{(K)} \qquad \begin{array}{r|ll} 1. & L & \text{P} \\ 2. & L \rightarrow (A \wedge K) & \text{P} \\ 3. & (A \wedge K) \rightarrow (L \rightarrow P) & \text{P} \\ \hline 4. & A \wedge K & 2,1 \rightarrow\text{E} \\ 5. & L \rightarrow P & 3,4 \rightarrow\text{E} \\ 6. & P & 5,1 \rightarrow\text{E} \end{array}$$

$L \rightarrow (A \wedge K)$ and L and are of the form $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$ where L is the $\mathcal{P}$ and $A \wedge K$ is $\mathcal{Q}$. So we use them to conclude $A \wedge K$ by $\rightarrow$E on (4). But then $(A \wedge K) \rightarrow (L \rightarrow P)$ and $A \wedge K$ are of the form $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$, so we use them to conclude $\mathcal{Q}$, in this

[2]I- and E-rules are often called *introduction* and *elimination* rules. This can lead to confusion as E-rules do not necessarily eliminate anything. The above, which is becoming more common, is more clear.

case, $L \to P$, on line (5). Finally $L \to P$ and L are of the form $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$, and we use them to conclude P on (6). Notice that,

(L)

$$
\begin{array}{r|lll}
1. & (A \to B) \wedge C & \text{P} & \\
2. & A & \text{P} & \\
\hline
3. & B & 1,2 \to\text{E} & \text{Mistake!}
\end{array}
$$

misapplies the rule. $(A \to B) \wedge C$ is not of the form $\mathcal{P} \to \mathcal{Q}$ — the main operator being $\wedge$, so that the formula is of the form $\mathcal{P} \wedge \mathcal{Q}$. The rule $\to$E applies just to formulas with main operator $\to$. If we want to use $(A \to B) \wedge C$ with A to conclude B, we would first have to isolate $A \to B$ on a line of its own. We might have done this in $N1$. But there is no rule for this (yet) in *ND*!

$\to$I is our first rule that requires a subderivation. Once we understand this rule, the rest are mere variations on a theme. $\to$I takes as its input an entire subderivation. Given an accessible subderivation which begins with assumption $\mathcal{P}$ on line *a* and ends with $\mathcal{Q}$ against the assumption's scope line at *b*, one may conclude $\mathcal{P} \to \mathcal{Q}$ with justification *a-b* $\to$I.

$\to$I

$$
\begin{array}{r||ll}
\text{a.} & \mathcal{P} & \text{A } (\mathcal{Q}, \to\text{I}) \\
 & & \\
\text{b.} & \mathcal{Q} & \\
 & \mathcal{P} \to \mathcal{Q} & \text{a-b } \to\text{I}
\end{array}
\quad \text{or} \quad
\begin{array}{r||ll}
\text{a.} & \mathcal{P} & \text{A } (g, \to\text{I}) \\
 & & \\
\text{b.} & \mathcal{Q} & \\
 & \mathcal{P} \to \mathcal{Q} & \text{a-b } \to\text{I}
\end{array}
$$

Note that the auxiliary assumption comes with a stated *exit strategy*: In this case the exit strategy includes the *formula* $\mathcal{Q}$ with which the subderivation is to end, and an indication of the rule ($\to$I) by which exit is to be made. We might write out the entire formula inside the parentheses as on the left. In practice, however, this is tedious, and it is easier just to write the formula at the bottom of the scope line where we will need it in the end. Thus in the parentheses on the right 'g' is a simple *pointer* to the goal formula at the end of the scope line. Note that the pointer is empty unless there is a formula to which it points, and *the exit strategy therefore is not complete unless the goal formula is stated*. In this case, the strategy includes the pointer to the goal formula, along with the indication of the rule ($\to$I) by which exit is to be made. Again, at the time we make the assumption, we write the $\mathcal{Q}$ down as part of the strategy for exiting the subderivation. But this does not mean the $\mathcal{Q}$ is justified! The $\mathcal{Q}$ is rather introduced as a new goal. Notice also that the justification *a-b* $\to$I does not refer to the *formulas* on lines *a* and *b*. These are inaccessible. Rather, the justification appeals to the subderivation which begins on line *a* and ends on line *b* — where this subderivation is accessible even though the formulas in it are not. So there is a difference between the comma and the hyphen, as they appear in justifications.

For this rule, we assume the antecedent, reach the consequent, and conclude to the conditional by →I. Intuitively, if an assumption $\mathcal{P}$ leads to $\mathcal{Q}$ then we know that *if* $\mathcal{P}$ *then* $\mathcal{Q}$. On truth tables, if there is a sententially valid argument from some other premises together with assumption $\mathcal{P}$ to conclusion $\mathcal{Q}$, then there is no row where those other premises are true and the assumption $\mathcal{P}$ is true but $\mathcal{Q}$ is false — but this is just to say that there is no row where the other premises are true and $\mathcal{P} \to \mathcal{Q}$ is false. We will have much more to say about this in Part III.

For an example, suppose we are confronted with the following.

(M)

1.	$A \to B$	P
2.	$B \to C$	P
	$A \to C$	

In general, we use an introduction rule to *produce* some formula — typically one already given as a goal. →I generates $\mathcal{P} \to \mathcal{Q}$ given a subderivation that starts with the $\mathcal{P}$ and ends with the $\mathcal{Q}$. Thus to reach $A \to C$, we need a subderivation that starts with A and ends with C. So we set up to reach $A \to C$ with the assumption A and an exit strategy to produce $A \to C$ by →I. For this we set the consequent C as a subgoal.

1.	$A \to B$	P
2.	$B \to C$	P
3.	$\quad A$	A (g, →I)
	$\quad C$	
	$A \to C$	

Again, we have not yet reached C or $A \to C$. Rather, we have assumed A and set C as a subgoal, with the strategy of terminating our subderivation by an application of →I. This much is stated in the exit strategy. As it happens, C is easy to get.

1.	$A \to B$	P
2.	$B \to C$	P
3.	$\quad A$	A (g, →I)
4.	$\quad B$	1,3 →E
5.	$\quad C$	2,4 →E
	$A \to C$	

Having reached C, and so completed the subderivation, we are in a position to execute our exit strategy and conclude $A \to C$ by →I.

1.	$A \to B$	P
2.	$B \to C$	P
3.	$\quad A$	A (g, $\to$I)
4.	$\quad B$	1,3 $\to$E
5.	$\quad C$	2,4 $\to$E
6.	$A \to C$	3-5 $\to$I

We appeal to the subderivation that starts with the assumption of the antecedent, and reaches the consequent. Notice that the $\to$I setup is driven, not by available premises and assumptions, but by where we want to get. We will say something more systematic about strategy once we have introduced all the rules. But here is the fundamental idea: *think goal directedly*. We begin with $A \to C$ as a goal. Our idea for producing it leads to C as a new goal. And the new goal is relatively easy to obtain.

Here is another example, one that should illustrate the above point about strategy, as well as the rule. Say we want to show $A \vdash_{ND} B \to (C \to A)$.

(N)

1.	A	P
	$B \to (C \to A)$	

Forget about the premise! Since the goal is of the form $\mathcal{P} \to \mathcal{Q}$, we set up to get it by $\to$I.

1.	A	P
2.	$\quad B$	A (g, $\to$I)
	$\quad C \to A$	
	$B \to (C \to A)$	

We need a subderivation that starts with the antecedent, and ends with the consequent. So we assume the antecedent, and set the consequent as a new goal. In this case, the new goal $C \to A$ has main operator $\to$, so we set up again to reach *it* by $\to$I.

1.	A	P
2.	$\quad B$	A (g, $\to$I)
3.	$\quad\quad C$	A (g, $\to$I)
	$\quad\quad A$	
	$\quad C \to A$	
	$B \to (C \to A)$	

The pointer g in an exit strategy points to the goal formula at the bottom of its scope line. Thus g for assumption B at (2) points to $C \rightarrow A$ at the bottom of its line, and g for assumption C at (3) points to A at the bottom of *its* line. Again, for the conditional, we assume the antecedent, and set the consequent as a new goal. And this last goal is particularly easy to reach. It follows immediately by reiteration from (1). Then it is a simple matter of executing the exit strategies with which our auxiliary assumptions were introduced.

1.	A	P
2.	$\quad B$	A (g, $\rightarrow$I)
3.	$\quad\quad C$	A (g, $\rightarrow$I)
4.	$\quad\quad A$	1 R
5.	$\quad C \rightarrow A$	3-4 $\rightarrow$I
6.	$B \rightarrow (C \rightarrow A)$	2-5 $\rightarrow$I

The subderivation which begins on (3) and ends on (4) begins with the antecedent and ends with the consequent of $C \rightarrow A$. So we conclude $C \rightarrow A$ on (5) by 3-4 $\rightarrow$I. The subderivation which begins on (2) and ends at (5) begins with the antecedent and ends with the consequent of $B \rightarrow (C \rightarrow A)$. So we reach $B \rightarrow (C \rightarrow A)$ on (6) by 2-5 $\rightarrow$I. Notice again how our overall reasoning is driven by the goals, rather than the premises and assumptions. It is sometimes difficult to motivate strategy when derivations are short and relatively easy. But this sort of thinking will stand you in good stead as problems get more difficult!

Given what we have done, the E- and I- rules for $\wedge$ are completely straightforward. If $\mathcal{P} \wedge \mathcal{Q}$ appears on some accessible line a of a derivation, then you may move to the $\mathcal{P}$, or to the $\mathcal{Q}$ with justification a $\wedge$E.

$\wedge$E

a.	$\mathcal{P} \wedge \mathcal{Q}$		a.	$\mathcal{P} \wedge \mathcal{Q}$	
	$\mathcal{P}$	a $\wedge$E		$\mathcal{Q}$	a $\wedge$E

Either qualifies as an instance of the rule. The left-hand case was R3 from $N1$. Intuitively, $\wedge$E should be clear. If $\mathcal{P}$ *and* $\mathcal{Q}$ is true, then $\mathcal{P}$ is true. And if $\mathcal{P}$ *and* $\mathcal{Q}$ is true, then $\mathcal{Q}$ is true. We saw a table for the left-hand case in (D). The other is similar. The $\wedge$ introduction rule is equally straightforward. If $\mathcal{P}$ and $\mathcal{Q}$ appear on accessible lines a and b of a derivation, then you may move to $\mathcal{P} \wedge \mathcal{Q}$ with justification *a,b* $\wedge$I.

$\wedge$I

a.	$\mathcal{P}$	
b.	$\mathcal{Q}$	
	$\mathcal{P} \wedge \mathcal{Q}$	a,b $\wedge$I

The order in which $\mathcal{P}$ and $\mathcal{Q}$ appear is irrelevant, though you should cite them in the specified order, line for the left conjunct first, and then for the right. If $\mathcal{P}$ is true and $\mathcal{Q}$ is true, then $\mathcal{P}$ *and* $\mathcal{Q}$ is true. Similarly, on a table, any line with both $\mathcal{P}$ and $\mathcal{Q}$ true has $\mathcal{P} \wedge \mathcal{Q}$ true.

Here is a simple example, demonstrating the *associativity* of conjunction.

(O)

1.	$A \wedge (B \wedge C)$	P
2.	A	1 $\wedge$E
3.	$B \wedge C$	1 $\wedge$E
4.	B	3 $\wedge$E
5.	C	3 $\wedge$E
6.	$A \wedge B$	2,4 $\wedge$I
7.	$(A \wedge B) \wedge C$	6,5 $\wedge$I

Notice that we could not get the B alone or the C alone without first isolating $B \wedge C$ on (3). As before, our rules apply just to the *main* operator. In effect, we take apart the premise with the E-rule, and put the conclusion together with the I-rule. Of course, as with $\rightarrow$I and $\rightarrow$E, rules for other operators do not always let us get to the parts and put them together in this simple and symmetric way.

Words to the wise:

- A common mistake made by beginning students is to assimilate other rules to $\wedge$E and $\wedge$I — moving, say, from $\mathcal{P} \rightarrow \mathcal{Q}$ alone to $\mathcal{P}$ or $\mathcal{Q}$, or from $\mathcal{P}$ and $\mathcal{Q}$ to $\mathcal{P} \rightarrow \mathcal{Q}$. *Do not forget what you have learned! Do not make this mistake!* The $\wedge$ rules are particularly easy. But each operator has its own special character. Thus $\rightarrow$E requires two "cards" to play. And $\rightarrow$I takes a subderivation as input.

- Another common mistake is to assume a formula $\mathcal{P}$ merely because it would be nice to have access to $\mathcal{P}$. *Do not make this mistake!* An assumption always comes with an exit strategy, and is useful only for application of the exit rule. At this stage, then, the *only* reason to assume $\mathcal{P}$ is to produce a formula of the sort $\mathcal{P} \rightarrow \mathcal{Q}$ by $\rightarrow$I.

A final example brings together all of the rules so far (except R).

(P)

$$\begin{array}{r|l|l}
1. & A \to C & \text{P} \\
\hline
2. & \quad A \wedge B & \text{A } (g, \to\text{I}) \\
3. & \quad A & 2 \wedge\text{E} \\
4. & \quad C & 1,3 \to\text{E} \\
5. & \quad B & 2 \wedge\text{E} \\
6. & \quad B \wedge C & 5,4 \wedge\text{I} \\
7. & (A \wedge B) \to (B \wedge C) & \text{2-6} \to\text{I}
\end{array}$$

We set up to obtain the overall goal by $\to$I. This generates $B \wedge C$ as a subgoal. We get $B \wedge C$ by getting the B and the C. Here is our guiding idea for strategy (which may now seem obvious): As you focus on a goal, to generate a formula with main operator $\star$, consider producing it by $\star$I. Thus, if the main operator of a goal or subgoal is $\to$, consider producing the formula by $\to$I; if the main operator of a goal is $\wedge$, consider producing it by $\wedge$I. This much should be sufficient for you to approach the following exercises. As you do the derivations, it is good simply to leave plenty of space on the page for your derivation as you state goal formulas, and let there be blank lines if room remains.[3]

E6.5. Complete the following derivations by filling in justifications for each line. Hint: it may be convenient to xerox the problems, and fill in your answers directly on the copy.

a.
$$\begin{array}{r|l}
1. & (A \wedge B) \to C \\
2. & B \wedge A \\
\hline
3. & B \\
4. & A \\
5. & A \wedge B \\
6. & C
\end{array}$$

b.
$$\begin{array}{r|l}
1. & (R \to L) \wedge [(S \vee R) \to (T \leftrightarrow K)] \\
2. & (R \to L) \to (S \vee R) \\
\hline
3. & R \to L \\
4. & S \vee R \\
5. & (S \vee R) \to (T \leftrightarrow K) \\
6. & T \leftrightarrow K
\end{array}$$

[3]Typing on a computer, it is easy to push lines down if you need more room. It is not so easy with pencil and paper, and worse with pen! If you decide to type, most word processors have a symbol font, with the capability of assigning symbols to particular keys. Assigning keys is far more efficient than finding characters over and over in menus.

c.

1.	B	
2.	$(A \to B) \to (B \to (L \wedge S))$	
3.	$\quad A$	
4.	$\quad B$	
5.	$A \to B$	
6.	$B \to (L \wedge S)$	
7.	$L \wedge S$	
8.	S	
9.	L	
10.	$S \wedge L$	

d.

1.	$A \wedge B$	
2.	$\quad C$	
3.	$\quad A$	
4.	$\quad A \wedge C$	
5.	$C \to (A \wedge C)$	
6.	$\quad C$	
7.	$\quad B$	
8.	$\quad B \wedge C$	
9.	$C \to (B \wedge C)$	
10.	$[C \to (A \wedge C)] \wedge [C \to (B \wedge C)]$	

e.

1.	$(A \wedge S) \to C$	
2.	$\quad A$	
3.	$\quad\quad S$	
4.	$\quad\quad A \wedge S$	
5.	$\quad\quad C$	
6.	$\quad S \to C$	
7.	$A \to (S \to C)$	

E6.6. The following are not legitimate *ND* derivations. In each case, explain why.

*a.

1.	$(A \wedge B) \wedge (C \to B)$	P
2.	A	1 $\wedge$E

b.

1.	$(A \wedge B) \wedge (C \to A)$	P
2.	C	P
3.	A	1,2 $\to$E

c.

1.	$(A \wedge B) \wedge (C \to A)$	P
2.	$C \to A$	1 $\wedge$E
3.	A	2 $\to$E

d.

1.	$A \to B$	P
2.	$\quad A \wedge C$	A ($g, \to$I)
3.	$\quad A$	2 $\wedge$E
4.	B	1,3 $\to$E

e.

1.	$A \to B$	P
2.	$\quad A \wedge C$	A ($g, \to$I)
3.	$\quad A$	2 $\wedge$E
4.	$\quad B$	1,3 $\to$E
5.	$\quad C$	2 $\wedge$E
6.	$\quad A \wedge C$	3,5 $\wedge$I

Hint: For this problem, think carefully about the exit strategy and the scope lines. Do we have the conclusion where we want it?

E6.7. Provide derivations to show each of the following.

a. $A \wedge B \vdash_{ND} B \wedge A$

*b. $A \wedge B, B \to C \vdash_{ND} C$

c. $A \wedge (A \to (A \wedge B)) \vdash_{ND} B$

d. $A \wedge B, B \to (C \wedge D) \vdash_{ND} A \wedge D$

*e. $A \to (A \to B) \vdash_{ND} A \to B$

f. $A, (A \wedge B) \to (C \wedge D) \vdash_{ND} B \to C$

g. $C \to A, C \to (A \to B) \vdash_{ND} C \to (A \wedge B)$

*h. $A \to B, B \to C \vdash_{ND} (A \wedge K) \to C$

i. $A \to B \vdash_{ND} (A \wedge C) \to (B \wedge C)$

j. $D \wedge E, (D \to F) \wedge (E \to G) \vdash_{ND} F \wedge G$

k. $O \to B, B \to S, S \to L \vdash_{ND} O \to L$

*l. $A \to B \vdash_{ND} (C \to A) \to (C \to B)$

m. $A \to (B \to C) \vdash_{ND} B \to (A \to C)$

n. $A \to (B \to C), D \to B \vdash_{ND} A \to (D \to C)$

o. $A \to B \vdash_{ND} A \to (C \to B)$

6.2.2 $\sim$ and $\vee$

Now let us consider the I- and E-rules for $\sim$ and $\vee$. The two rules for $\sim$ are quite similar to one another. Each appeals to a single subderivation. For $\sim$I, given an accessible subderivation which begins with assumption $\mathcal{P}$ on line a, and ends with a formula of the form $\mathcal{Q} \wedge \sim\mathcal{Q}$ against its scope line on line b, one may conclude $\sim\mathcal{P}$ by a-b $\sim$I. For $\sim$E, given an accessible subderivation which begins with assumption $\sim\mathcal{P}$ on line a, and ends with a formula of the form $\mathcal{Q} \wedge \sim\mathcal{Q}$ against its scope line on line b, one may conclude $\mathcal{P}$ by a-b $\sim$E.

$\sim$I	a.	$\mathcal{P}$	A (c, $\sim$I)	$\sim$E	a.	$\sim\mathcal{P}$	A (c, $\sim$E)
	b.	$\mathcal{Q} \wedge \sim\mathcal{Q}$			b.	$\mathcal{Q} \wedge \sim\mathcal{Q}$	
		$\sim\mathcal{P}$	a-b $\sim$I			$\mathcal{P}$	a-b $\sim$E

$\sim$I introduces an expression with main operator tilde, adding tilde to the assumption $\mathcal{P}$. $\sim$E exploits the assumption $\sim\mathcal{P}$, with a result that takes the tilde off. For these rules, the formula $\mathcal{Q}$ may be *any* formula, so long as $\sim\mathcal{Q}$ is *it* with a tilde in front. Because $\mathcal{Q}$ may be any formula, when we declare our exit strategy for the assumption, we might have no particular goal formula in mind. So, where g always points to a formula written at the bottom of a scope line, c is not a pointer to any particular formula. Rather, when we declare our exit strategy, we merely indicate our intent to obtain some contradiction, and then to exit by $\sim$I or $\sim$E.

Intuitively, if an assumption leads to a result that is false, the assumption is wrong. So if the assumption $\mathcal{P}$ leads to $\mathcal{Q} \wedge \sim\mathcal{Q}$, then $\sim\mathcal{P}$; and if the assumption $\sim\mathcal{P}$ leads to $\mathcal{Q} \wedge \sim\mathcal{Q}$, then $\mathcal{P}$. On tables, there can be no row where $\mathcal{Q} \wedge \sim\mathcal{Q}$ is true; so if every row where some premises together with assumption $\mathcal{P}$ are true would have to make both $\mathcal{Q} \wedge \sim\mathcal{Q}$ true, then there can be no row where those other premises are true and $\mathcal{P}$ is true — so any row where the other premises are true is one where $\mathcal{P}$ is false, and $\sim\mathcal{P}$ is therefore true. Similarly when the assumption is $\sim\mathcal{P}$, any row where the other premises are true has to be one where $\sim\mathcal{P}$ is false, so that $\mathcal{P}$ is true. Again, we will have much more to say about this reasoning in Part III.

Here are some examples of these rules. Notice that, again, we introduce subderivations with the overall goal in mind.

(Q)

1.	$A \rightarrow B$	P
2.	$A \rightarrow \sim B$	P
3.	$\quad A$	A (c, $\sim$I)
4.	$\quad B$	1,3 $\rightarrow$E
5.	$\quad \sim B$	2,3 $\rightarrow$E
6.	$\quad B \wedge \sim B$	4,5 $\wedge$I
7.	$\sim A$	3-6, $\sim$I

We begin with the goal of obtaining $\sim A$. The natural way to obtain this is by $\sim$I. So we set up a subderivation with that in mind. Since the goal is $\sim A$, we begin with A, and go for a contradiction. In this case, the contradiction is easy to obtain, by a couple applications of $\rightarrow$E and then $\wedge$I.

Here is another case that may be more interesting.

(R)

1.	$\sim A$	P
2.	$B \rightarrow A$	P
3.	$\quad L \wedge B$	A (c, $\sim$I)
4.	$\quad B$	3 $\wedge$E
5.	$\quad A$	2,4 $\rightarrow$E
6.	$\quad A \wedge \sim A$	5,1 $\wedge$I
7.	$\sim(L \wedge B)$	3-6 $\sim$I

This time, the original goal is $\sim(L \wedge B)$. It is of the form $\sim\mathcal{P}$, so we set up to obtain it with a subderivation that begins with the $\mathcal{P}$, that is, $L \wedge B$. In this case, the contradiction is $A \wedge \sim A$. Once we have the contradiction, we simply apply our exit strategy.

A simplification. Let $\perp$ (bottom) abbreviate an arbitrary contradiction — say $Z \wedge \sim Z$. Adopt a rule $\perp$I as on the left below,

$\perp$I

a.	$\mathcal{Q}$	
b.	$\sim Q$	
	$\perp$	a,b $\perp$I

(S)

1.	$\mathcal{Q}$	
2.	$\sim Q$	
3.	$\quad \sim(Z \wedge \sim Z)$	A (c $\sim$E)
4.	$\quad \mathcal{Q} \wedge \sim\mathcal{Q}$	1,2 $\wedge$I
5.	$Z \wedge \sim Z$	3-4 $\sim$E

Given $\mathcal{Q}$ and $\sim\mathcal{Q}$ on accessible lines, we move directly to $\perp$ by $\perp$I. This is an example of a *derived* rule. For, given $\mathcal{Q}$ and $\sim\mathcal{Q}$, we can always derive $Z \wedge \sim Z$ (that is, $\perp$) as in (S) on the right. Given this, the $\sim$I and $\sim$E rules appear in the forms,

$\sim$I

a.	$\mid\ \mid \mathcal{P}$	A (c, $\sim$I)
b.	$\mid\ \mid \bot$	
	$\mid \sim\mathcal{P}$	a-b $\sim$I

$\sim$E

a.	$\mid\ \mid \sim\mathcal{P}$	A (c, $\sim$E)
b.	$\mid\ \mid \bot$	
	$\mid \mathcal{P}$	a-b $\sim$E

Since $\bot$ is (abbreviates) a sentence of the form $\mathcal{Q}\wedge\sim\mathcal{Q}$, the subderivations for $\sim$I and $\sim$E are appropriately concluded with $\bot$. Observe that with $\bot$ at the bottom the $\sim$I and $\sim$E rules have a particular goal sentence, very much like $\rightarrow$I. However, the $\mathcal{Q}$ and $\sim\mathcal{Q}$ required to obtain $\bot$ by $\bot$I are the same as would be required for $\mathcal{Q}\wedge\sim\mathcal{Q}$ on the original form of the rules. For this reason, we declare our exit strategy with a c rather than g any time the goal is $\bot$. At one level, this simplification is a mere notational convenience: having obtained $\mathcal{Q}$ and $\sim\mathcal{Q}$, we move to $\bot$, instead of writing out the complex conjunction $\mathcal{Q}\wedge\sim\mathcal{Q}$. However, there are contexts where it will be convenient to have a *particular* contradiction as goal. Thus this is the standard form in which we use these rules.

Here is an example of the rules in this form, this time for $\sim$E.

(T)

1.	$\mid \sim\sim A$	P
2.	$\mid\ \mid \sim A$	A (c, $\sim$E)
3.	$\mid\ \mid \bot$	2,1 $\bot$I
4.	$\mid A$	2-3 $\sim$E

It is no surprise that we can derive A from $\sim\sim A$! This is how to do it in *ND*. Again, do not begin by thinking about the premise. The goal is A, and we can get it with a subderivation that starts with $\sim A$, by a $\sim$E exit strategy. In this case the $\mathcal{Q}$ and $\sim\mathcal{Q}$ for $\bot$I are $\sim A$ and $\sim\sim A$ — that is $\sim A$ with a tilde in front of it. Though very often (at least in the beginning) an atomic and its negation will do for your contradiction, $\mathcal{Q}$ and $\sim\mathcal{Q}$ need not be simple. Observe that $\sim$E is a strange and powerful rule: Though an E-rule, effectively it can be used in pursuit of any goal whatsoever — to obtain formula $\mathcal{P}$ by $\sim$E, all one has to do is obtain a contradiction from the assumption of $\mathcal{P}$ with a tilde in front. As in this last example (T), $\sim$E is particularly useful when the goal is an atomic formula, and thus without a main operator, so that there is no straightforward way for regular introduction rules to apply. In this way, it plays the role of a sort of "back door" introduction role.

The $\vee$I and and $\vee$E rules apply methods we have already seen. For $\vee$I, given an accessible formula $\mathcal{P}$ on line a, one may move to either $\mathcal{P}\vee\mathcal{Q}$ or to $\mathcal{Q}\vee\mathcal{P}$ for any formula $\mathcal{Q}$, with justification a $\vee$I.

$$\vee\text{I}\qquad \begin{array}{r|ll} \text{a.} & \mathcal{P} & \\ & & \\ & \mathcal{P}\vee\mathcal{Q} & \text{a }\vee\text{I} \end{array} \qquad\qquad \begin{array}{r|ll} \text{a.} & \mathcal{P} & \\ & & \\ & \mathcal{Q}\vee\mathcal{P} & \text{a }\vee\text{I} \end{array}$$

The left-hand case was R4 from $N1$. Also, we saw an intuitive version of this rule as *addition* on p. 26. Table (D) exhibits the left-hand case. And the other side should be clear as well: Any row of a table where $\mathcal{P}$ is true has both $\mathcal{P} \vee \mathcal{Q}$ and $\mathcal{Q} \vee \mathcal{P}$ true.

Here is a simple example.

$$\text{(U)}\qquad \begin{array}{r|ll} 1. & P & \text{P} \\ 2. & \underline{(P \vee Q) \rightarrow R} & \text{P} \\ 3. & P \vee Q & 1\ \vee\text{I} \\ 4. & R & 2,3\ \rightarrow\text{E} \end{array}$$

It is easy to get R once we have $P \vee Q$. And we build $P \vee Q$ directly from the P. Note that we could have done the derivation as well if (2) had been, say, $(P \vee [K \wedge (L \leftrightarrow T)]) \rightarrow R$ and we used $\vee$I to add $[K \wedge (L \leftrightarrow T)]$ to the P all at once.

The inputs to $\vee$E are a formula of the form $\mathcal{P} \vee \mathcal{Q}$ and *two* subderivations. Given an accessible formula of the form $\mathcal{P} \vee \mathcal{Q}$ on line a, with an accessible subderivation beginning with assumption $\mathcal{P}$ on line b and ending with conclusion $\mathcal{C}$ against its scope line at c, and an accessible subderivation beginning with assumption $\mathcal{Q}$ on line d and ending with conclusion $\mathcal{C}$ against its scope line at e, one may conclude $\mathcal{C}$ with justification *a,b-c,d-e* $\vee$E.

$$\vee\text{E}\qquad \begin{array}{r|l|ll} \text{a.} & \multicolumn{2}{l}{\mathcal{P}\vee\mathcal{Q}} & \\ \text{b.} & & \underline{\mathcal{P}} & \text{A } (g,\ \text{a}\vee\text{E}) \\ & & & \\ \text{c.} & & \mathcal{C} & \\ \text{d.} & & \underline{\mathcal{Q}} & \text{A } (g,\ \text{a}\vee\text{E}) \\ & & & \\ \text{e.} & & \mathcal{C} & \\ & \multicolumn{2}{l}{\mathcal{C}} & \text{a,b-c,d-e }\vee\text{E} \end{array}$$

Given a disjunction $\mathcal{P} \vee \mathcal{Q}$, one subderivation begins with $\mathcal{P}$, and the other with $\mathcal{Q}$; both concluding with $\mathcal{C}$. This time our exit strategy includes markers for the new subgoals, along with a notation that we exit by appeal to the disjunction on line a and $\vee$E. Intuitively, if we know it is one or the other, and *either* leads to some conclusion, then the conclusion must be true. Here is an example a student gave me near graduation time: She and her mother were shopping for a graduation dress. They narrowed it down to dress A or dress B. Dress A was expensive, and if they bought it, her mother would be mad. But dress B was ugly and if they bought it

the student would complain and her mother would be mad. Conclusion: her mother would be mad — and this without knowing which dress they were going to buy! On a truth table, if rows where $\mathcal{P}$ is true have $\mathcal{C}$ true, and rows where $\mathcal{Q}$ is true have $\mathcal{C}$ true, then any row with $\mathcal{P} \vee \mathcal{Q}$ true must have $\mathcal{C}$ true as well.

Here are a couple of examples. The first is straightforward, and illustrates both the $\vee$I and $\vee$E rules.

(V)

1.	$A \vee B$	P
2.	$A \to C$	P
3.	$\quad A$	A (g, 1$\vee$E)
4.	$\quad C$	2,3 $\to$E
5.	$\quad B \vee C$	4 $\vee$I
6.	$\quad B$	A (g, 1$\vee$E)
7.	$\quad B \vee C$	6 $\vee$I
8.	$B \vee C$	1,3-5,6-7 $\vee$E

We have the disjunction $A \vee B$ as premise, and original goal $B \vee C$. And we set up to obtain the goal by $\vee$E. For this, one subderivation starts with A and ends with $B \vee C$, and the other starts with B and ends with $B \vee C$. As it happens, these subderivations are easy to complete.

Very often, beginning students resist using $\vee$E — no doubt because it is relatively messy. *But this is a mistake — $\vee$E is your friend*! In fact, with this rule, we have a case where it pays to look at the premises for general strategy. Again, we will have more to say later. But if you have a premise or accessible line of the form $\mathcal{P} \vee \mathcal{Q}$, you should go for your goal, whatever it is, by $\vee$E. Here is why: As you go for the goal in the first subderivation, you have whatever premises were accessible before, *plus* $\mathcal{P}$; and as you go for the goal in the second subderivation, you have whatever premises were accessible before *plus* $\mathcal{Q}$. So you can only be better off in your quest to reach the goal. In many cases where a premise has main operator $\vee$, there is no way to complete the derivation except by $\vee$E. The above example (V) is a case in point.

Here is a relatively messy example, which should help you be sure you understand the $\vee$ rules. It illustrates the *associativity* of disjunction.

(W)

1.	$A \vee (B \vee C)$	P
2.	$\quad A$	A (g, 1$\vee$E)
3.	$\quad A \vee B$	2 $\vee$I
4.	$\quad (A \vee B) \vee C$	3 $\vee$I
5.	$\quad B \vee C$	A (g, 1$\vee$E)
6.	$\quad\quad B$	A (g, 5$\vee$E)
7.	$\quad\quad A \vee B$	6 $\vee$I
8.	$\quad\quad (A \vee B) \vee C$	7 $\vee$I
9.	$\quad\quad C$	A (g, 5$\vee$E)
10.	$\quad\quad (A \vee B) \vee C$	9 $\vee$I
11.	$\quad (A \vee B) \vee C$	5,6-8,9-10 $\vee$E
12.	$(A \vee B) \vee C$	1,2-4,5-11 $\vee$E

The premise has main operator $\vee$. So we set up to obtain the goal by $\vee$E. This gives us subderivations starting with A and $B \vee C$, each with $(A \vee B) \vee C$ as goal. The first is easy to complete by a couple instances of $\vee$I. But the assumption of the second, $B \vee C$ has main operator $\vee$. So we set up to obtain *its* goal by $\vee$E. This gives us subderivations starting with B and C, each again having $(A \vee B) \vee C$ as goal. Again, these are easy to complete by application of $\vee$I. The final result follows by the planned applications of $\vee$E. If you have been able to follow this case, you are doing well!

E6.8. Complete the following derivations by filling in justifications for each line.

a.

1.	$\sim B$
2.	$(\sim A \vee C) \to (B \wedge C)$
3.	$\quad \sim A$
4.	$\quad \sim A \vee C$
5.	$\quad B \wedge C$
6.	$\quad B$
7.	$\quad \perp$
8.	A

b.

1. $\mid R$
2. $\mid {\sim}(S \vee T)$
3. $\mid \mid R \rightarrow S$
4. $\mid \mid S$
5. $\mid \mid S \vee T$
6. $\mid \mid \bot$
7. $\mid {\sim}(R \rightarrow S)$

c.

1. $\mid (R \wedge S) \vee (K \wedge L)$
2. $\mid \mid R \wedge S$
3. $\mid \mid R$
4. $\mid \mid S$
5. $\mid \mid S \wedge R$
6. $\mid \mid (S \wedge R) \vee (L \wedge K)$
7. $\mid \mid K \wedge L$
8. $\mid \mid K$
9. $\mid \mid L$
10. $\mid \mid L \wedge K$
11. $\mid \mid (S \wedge R) \vee (L \wedge K)$
12. $\mid (S \wedge R) \vee (L \wedge K)$

d.

1. $\mid A \vee B$
2. $\mid \mid A$
3. $\mid \mid \mid A \rightarrow B$
4. $\mid \mid \mid B$
5. $\mid \mid (A \rightarrow B) \rightarrow B$
6. $\mid \mid B$
7. $\mid \mid \mid A \rightarrow B$
8. $\mid \mid \mid B$
9. $\mid \mid (A \rightarrow B) \rightarrow B$
10. $\mid (A \rightarrow B) \rightarrow B$

e.

1.	$\sim B$
2.	$\sim A \rightarrow (A \vee B)$
3.	$\mid\ \sim A$
4.	$\mid\ A \vee B$
5.	$\mid\ \mid\ A$
6.	$\mid\ \mid\ A$
7.	$\mid\ \mid\ B$
8.	$\mid\ \mid\ \mid\ \sim A$
9.	$\mid\ \mid\ \mid\ \bot$
10.	$\mid\ \mid\ A$
11.	$\mid\ A$
12.	$\mid\ \bot$
13.	A

E6.9. The following are not legitimate *ND* derivations. In each case, explain why.

a.

1.	$A \vee B$	P
2.	B	1 $\vee$E

b.

1.	$\sim A$	P
2.	$B \rightarrow A$	P
3.	$\mid\ B$	A (c, $\sim$I)
4.	$\mid\ A$	2,3 $\rightarrow$E
5.	$\sim B$	3-4 $\sim$I

*c.

1.	W	P
2.	$\mid\ R$	A (c, $\sim$I)
3.	$\mid\ \mid\ \sim W$	A (c, $\sim$I)
4.	$\mid\ \mid\ \bot$	1,3 $\bot$I
5.	$\sim R$	2-4 $\sim$I

d.

1.	$A \vee B$	P
2.	$\mid A$	A (g, 1$\vee$E)
3.	$\mid A$	2 R
4.	$\mid B$	A (g, 1$\vee$E)
5.	$\mid A$	3 R
6.	A	1,2-3,4-5 $\vee$E

e.

1.	$A \vee B$	P
2.	$\mid A$	A (g, 1$\vee$E)
3.	$\mid A$	2 R
4.	$\mid A$	A (c, $\sim$I)
5.	$\mid \mid B$	A (g, 1$\vee$E)
6.	$\mid \mid A$	4 R
7.	A	1,2-3,5-6 $\vee$E

E6.10. Produce derivations to show each of the following.

a. $\sim A \vdash_{ND} \sim(A \wedge B)$

b. $A \vdash_{ND} \sim\sim A$

*c. $\sim A \rightarrow B, \sim B \vdash_{ND} A$

d. $A \rightarrow B \vdash_{ND} \sim(A \wedge \sim B)$

e. $\sim A \rightarrow B,\ B \rightarrow A \vdash_{ND} A$

f. $A \wedge B \vdash_{ND} (R \leftrightarrow S) \vee B$

*g. $A \vee (A \wedge B) \vdash_{ND} A$

h. $S, (B \vee C) \rightarrow \sim S \vdash_{ND} \sim B$

i. $A \vee B,\ A \rightarrow B,\ B \rightarrow A \vdash_{ND} A \wedge B$

j. $A \rightarrow B,\ (B \vee C) \rightarrow D,\ D \rightarrow \sim A \vdash_{ND} \sim A$

k. $A \vee B \vdash_{ND} B \vee A$

*l. $A \rightarrow \sim B \vdash_{ND} B \rightarrow \sim A$

m. $(A \wedge B) \rightarrow \sim A \vdash_{ND} A \rightarrow \sim B$

n. $A \vee \sim\sim B \vdash_{ND} A \vee B$

o. $A \vee B, \sim B \vdash_{ND} A$

6.2.3 $\leftrightarrow$

We complete our presentation of rules for the sentential part of *ND* with the rules $\leftrightarrow$E and $\leftrightarrow$I. Given that $\mathcal{P} \leftrightarrow \mathcal{Q}$ abbreviates the same as $(\mathcal{P} \rightarrow \mathcal{Q}) \wedge (\mathcal{Q} \rightarrow \mathcal{P})$, it is not surprising that rules for $\leftrightarrow$ work like ones for arrow, but going two ways. For $\leftrightarrow$E, if formulas $\mathcal{P} \leftrightarrow \mathcal{Q}$ and $\mathcal{P}$ appear on accessible lines a and b of a derivation, we may conclude $\mathcal{Q}$ with justification *a,b* $\leftrightarrow$E; and similarly but in the other direction, if formulas $\mathcal{P} \leftrightarrow \mathcal{Q}$ and $\mathcal{Q}$ appear on accessible lines a and b of a derivation, we may conclude $\mathcal{P}$ with justification *a,b* $\leftrightarrow$E.

$$
\leftrightarrow\text{E} \qquad
\begin{array}{r|ll}
\text{a.} & \mathcal{P} \leftrightarrow \mathcal{Q} & \\
\text{b.} & \mathcal{P} & \\
 & & \\
 & \mathcal{Q} & \text{a,b} \leftrightarrow\text{E}
\end{array}
\qquad\qquad
\begin{array}{r|ll}
\text{a.} & \mathcal{P} \leftrightarrow \mathcal{Q} & \\
\text{b.} & \mathcal{Q} & \\
 & & \\
 & \mathcal{P} & \text{a,b} \leftrightarrow\text{E}
\end{array}
$$

$\mathcal{P} \leftrightarrow \mathcal{Q}$ thus works like either $\mathcal{P} \rightarrow \mathcal{Q}$ or $\mathcal{Q} \rightarrow \mathcal{P}$. Intuitively given $\mathcal{P}$ if and *only if* $\mathcal{Q}$, then if $\mathcal{P}$ is true, $\mathcal{Q}$ is true. And given $\mathcal{P}$ *if* and only if $\mathcal{Q}$, then if $\mathcal{Q}$ is true $\mathcal{P}$ is true. On tables, if $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true, then $\mathcal{P}$ and $\mathcal{Q}$ have the same truth value. So if $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true and $\mathcal{P}$ is true, $\mathcal{Q}$ is true as well; and if $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true and $\mathcal{Q}$ is true, $\mathcal{P}$ is true as well.

Given that $\mathcal{P} \leftrightarrow \mathcal{Q}$ can be exploited like $\mathcal{P} \rightarrow \mathcal{Q}$ or $\mathcal{Q} \rightarrow \mathcal{P}$, it is not surprising that introducing $\mathcal{P} \leftrightarrow \mathcal{Q}$ is like introducing both $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{Q} \rightarrow \mathcal{P}$. The input to $\leftrightarrow$I is *two* subderivations. Given an accessible subderivation beginning with assumption $\mathcal{P}$ on line a and ending with conclusion $\mathcal{Q}$ against its scope line on b, and an accessible subderivation beginning with assumption $\mathcal{Q}$ on line c and ending with conclusion $\mathcal{P}$ against its scope line on d, one may conclude $\mathcal{P} \leftrightarrow \mathcal{Q}$ with justification, *a-b,c-d* $\leftrightarrow$I.

$$
\leftrightarrow\text{I} \qquad
\begin{array}{r|l|ll}
\text{a.} & & \mathcal{P} & \text{A } (g, \leftrightarrow\text{I}) \\
\cline{3-3}
 & & & \\
\text{b.} & & \mathcal{Q} & \\
 & & & \\
\text{c.} & & \mathcal{Q} & \text{A } (g, \leftrightarrow\text{I}) \\
\cline{3-3}
 & & & \\
\text{d.} & & \mathcal{P} & \\
 & \multicolumn{2}{l}{\mathcal{P} \leftrightarrow \mathcal{Q}} & \text{a-b,c-d} \leftrightarrow\text{I}
\end{array}
$$

Intuitively, if an assumption $\mathcal{P}$ leads to $\mathcal{Q}$ and the assumption $\mathcal{Q}$ leads to $\mathcal{P}$, then we know that *if* $\mathcal{P}$ then $\mathcal{Q}$, and *if* $\mathcal{Q}$ then $\mathcal{P}$ — which is to say that $\mathcal{P}$ if and only if $\mathcal{Q}$. On truth tables, if there is a sententially valid argument from some other premises together with assumption $\mathcal{P}$, to conclusion $\mathcal{Q}$, then there is no row where those other premises are true and assumption $\mathcal{P}$ is true and $\mathcal{Q}$ is false; and if there is a sententially valid argument from those other premises together with assumption $\mathcal{Q}$ to conclusion $\mathcal{P}$, then there is no row where those other premises are true and the assumption $\mathcal{Q}$ is true and $\mathcal{P}$ is false; so on rows where the other premises are true, $\mathcal{P}$ and $\mathcal{Q}$ do not have different values, and the biconditional $\mathcal{P} \leftrightarrow \mathcal{Q}$ is true.

Here are a couple of examples. The first is straightforward, and exercises both the $\leftrightarrow$I and $\leftrightarrow$E rules. We show, $A \leftrightarrow B, B \leftrightarrow C \vdash_{ND} A \leftrightarrow C$.

(X)

$$
\begin{array}{rlll}
1. & A \leftrightarrow B & & \text{P} \\
2. & B \leftrightarrow C & & \text{P} \\
3. & \quad A & & \text{A } (g, \leftrightarrow\text{I}) \\
4. & \quad B & & 1,3 \leftrightarrow\text{E} \\
5. & \quad C & & 2,4 \leftrightarrow\text{E} \\
6. & \quad C & & \text{A } (g, \leftrightarrow\text{I}) \\
7. & \quad B & & 2,6 \leftrightarrow\text{E} \\
8. & \quad A & & 1,7 \leftrightarrow\text{E} \\
9. & A \leftrightarrow C & & \text{3-5,6-8} \leftrightarrow\text{I}
\end{array}
$$

Our original goal is $A \leftrightarrow C$. So it is natural to set up subderivations to get it by $\leftrightarrow$I. Once we have done this, the subderivations are easily completed by applications of $\leftrightarrow$E.

Here is an interesting case that again exercises both rules. We show, $A \leftrightarrow (B \leftrightarrow C), C \vdash_{ND} A \leftrightarrow B$.

ND Quick Reference (Sentential)

R (*reiteration*)

a. | $\mathcal{P}$
| $\mathcal{P}$ a R

$\sim$I (*negation intro*)

a. | | $\mathcal{P}$ A (c, $\sim$I)
b. | | $\mathcal{Q} \wedge \sim\mathcal{Q}$ ($\bot$)
| $\sim\mathcal{P}$ a-b $\sim$I

$\sim$E (*negation exploit*)

a. | | $\sim\mathcal{P}$ A (c, $\sim$E)
b. | | $\mathcal{Q} \wedge \sim\mathcal{Q}$ ($\bot$)
| $\mathcal{P}$ a-b $\sim$E

$\wedge$I (*conjunction intro*)

a. | $\mathcal{P}$
b. | $\mathcal{Q}$
| $\mathcal{P} \wedge \mathcal{Q}$ a,b $\wedge$I

$\wedge$E (*conjunction exploit*)

a. | $\mathcal{P} \wedge \mathcal{Q}$
| $\mathcal{P}$ a $\wedge$E

$\wedge$E (*conjunction exploit*)

a. | $\mathcal{P} \wedge \mathcal{Q}$
| $\mathcal{Q}$ a $\wedge$E

$\vee$I (*disjunction intro*)

a. | $\mathcal{P}$
| $\mathcal{P} \vee \mathcal{Q}$ a $\vee$I

$\vee$I (*disjunction intro*)

a. | $\mathcal{P}$
| $\mathcal{Q} \vee \mathcal{P}$ a $\vee$I

$\vee$E (*disjunction exploit*)

a. | $\mathcal{P} \vee \mathcal{Q}$
b. | | $\mathcal{P}$ A (g, a$\vee$E)
c. | | $\mathcal{C}$
d. | | $\mathcal{Q}$ A (g, a$\vee$E)
e. | | $\mathcal{C}$
| $\mathcal{C}$ a,b-c,d-e $\vee$E

$\rightarrow$I (*conditional intro*)

a. | | $\mathcal{P}$ A (g, $\rightarrow$I)
b. | | $\mathcal{Q}$
| $\mathcal{P} \rightarrow \mathcal{Q}$ a-b $\rightarrow$I

$\rightarrow$E (*conditional exploit*)

a. | $\mathcal{P} \rightarrow \mathcal{Q}$
b. | $\mathcal{P}$
| $\mathcal{Q}$ a,b $\rightarrow$E

$\leftrightarrow$I (*biconditional intro*)

a. | | $\mathcal{P}$ A (g, $\leftrightarrow$I)
b. | | $\mathcal{Q}$
c. | | $\mathcal{Q}$ A (g, $\leftrightarrow$I)
d. | | $\mathcal{P}$
| $\mathcal{P} \leftrightarrow \mathcal{Q}$ a-b,c-d $\leftrightarrow$I

$\leftrightarrow$E (*biconditional exploit*)

a. | $\mathcal{P} \leftrightarrow \mathcal{Q}$
b. | $\mathcal{P}$
| $\mathcal{Q}$ a,b $\leftrightarrow$E

$\leftrightarrow$E (*biconditional exploit*)

a. | $\mathcal{P} \leftrightarrow \mathcal{Q}$
b. | $\mathcal{Q}$
| $\mathcal{P}$ a,b $\leftrightarrow$E

$\bot$I (*bottom intro*)

a. | $\mathcal{Q}$
b. | $\sim\mathcal{Q}$
| $\bot$ a,b $\bot$I

(Y)

1.	$A \leftrightarrow (B \leftrightarrow C)$	P
2.	C	P
3.	$\mid A$	A (g, $\leftrightarrow$I)
4.	$\mid B \leftrightarrow C$	1,3 $\leftrightarrow$E
5.	$\mid B$	4,2 $\leftrightarrow$E
6.	$\mid B$	A (g, $\leftrightarrow$I)
7.	$\mid \mid B$	A (g, $\leftrightarrow$I)
8.	$\mid \mid C$	2 R
9.	$\mid \mid C$	A (g, $\leftrightarrow$I)
10.	$\mid \mid B$	6 R
11.	$\mid B \leftrightarrow C$	7-8,9-10 $\leftrightarrow$I
12.	$\mid A$	1,11 $\leftrightarrow$E
13.	$A \leftrightarrow B$	3-5,6-12 $\leftrightarrow$I

We begin by setting up the subderivations to get $A \leftrightarrow B$ by $\leftrightarrow$I. This first is easily completed with a couple applications of $\leftrightarrow$E. To reach the goal for the second by means of the premise (1) we need $B \leftrightarrow C$ as our second "card." So we set up to reach *that*. As it happens, the extra subderivations at (7) - (8) and (9) - (10) are easy to complete. Again, if you have followed so far, you are doing well. We will be in a better position to *create* such derivations after our discussion of strategy.

So much for the rules for this sentential part of *ND*. Before we turn in the next sections to strategy, let us note a couple of features of the rules that may so-far have gone without notice. First, premises are not always necessary for *ND* derivations. Thus, for example, $\vdash_{ND} A \to A$.

(Z)

1.	$\mid A$	A (g, $\to$I)
2.	$\mid A$	1 R
3.	$A \to A$	1-2 $\to$I

If there are no premises, do not panic! Begin in the usual way. In this case, the original goal is $A \to A$. So we set up to obtain it by $\to$I. And the subderivation is particularly simple. Notice that our derivation of $A \to A$ corresponds to the fact from truth tables that $\vDash_s A \to A$. And we *need* to be able to derive $A \to A$ from no premises if there is to be the right sort of correspondence between derivations in *ND* and semantic validity — if we are to have $\Gamma \vDash \mathscr{P}$ iff $\Gamma \vdash_{ND} P$.

Second, observe again that every subderivation comes with an exit strategy. The exit strategy says whether you intend to complete the subderivation with a particular goal, or by obtaining a contradiction, and then how the subderivation is to be used

once complete. There are just five rules which appeal to a subderivation: →I, ∼I, ∼E, ∨E, and ↔I. You will complete the subderivation, and then use it by one of these rules. So these are the *only* rules which may appear in an exit strategy. If you do not understand this, then you need to go back and think about the rules until you do.

Finally, it is worth noting a strange sort of case, with application to rules that can take more than one input of the same type. Consider a simple demonstration that $A \vdash_{ND} A \wedge A$. We might proceed as in (AA) on the left,

(AA)
$$\begin{array}{r|lll} 1. & A & & \text{P} \\ \hline 2. & A & & 1\ \text{R} \\ 3. & A \wedge A & & 1,2\ \wedge\text{I} \end{array}$$

(AB)
$$\begin{array}{r|lll} 1. & A & & \text{P} \\ \hline 3. & A \wedge A & & 1,1\ \wedge\text{I} \end{array}$$

We begin with A, reiterate so that A appears on different lines, and apply ∧I. But we might have proceeded as in (AB) on the right. The rule requires an accessible line on which the left conjunct appears — which we have at (1), and an accessible line on which the right conjunct appears *which we also have* on (1). So the rule takes an input for the left conjunct and an input for the right — they just happen to be the same thing. A similar point applies to rules ∨E and ↔I which take more than one subderivation as input. Suppose we want to show $A \vee A \vdash_{ND} A$.[4]

(AC)
$$\begin{array}{r|l|ll} 1. & A \vee A & & \text{P} \\ \hline 2. & & A & \text{A } (g,\ 1\vee\text{E}) \\ \cline{3-3} 3. & & A & 2\ \text{R} \\ 4. & & A & \text{A } (g,\ 1\vee\text{E}) \\ \cline{3-3} 5. & & A & 4\ \text{R} \\ 6. & A & & 1,2\text{-}3,4\text{-}5\ \vee\text{E} \end{array}$$

(AD)
$$\begin{array}{r|l|ll} 1. & A \vee A & & \text{P} \\ \hline 2. & & A & \text{A } (g,\ 1\vee\text{E}) \\ \cline{3-3} 3. & & A & 2\ \text{R} \\ 4. & A & & 1,2\text{-}3,2\text{-}3\ \vee\text{E} \end{array}$$

In (AC), we begin in the usual way to get the main goal by ∨E. This leads to the subderivations (2) - (3) and (4) - (5), the first moving from the left disjunct to the goal, and the second from the right disjunct to the goal. But the left and right disjuncts are the same! So we might have simplified as in (AD). ∨E still requires three inputs: First an accessible disjunction, which we find on (1); second an accessible subderivation which moves from the left disjunct to the goal, which we find on (2) - (3); third a subderivation which moves from the right disjunct to the goal — *but we have this on* (2) - (3). So the justification at (4) of (AD) appeals to the three relevant facts, by appeal to the same subderivation twice. Similarly one could imagine a quick-and-dirty demonstration that $\vdash_{ND} A \leftrightarrow A$.

[4] I am reminded of an irritating character in *Groundhog Day* who repeatedly asks, "Am I right or am I right?" If he implies that the disjunction is true, it follows that he is right.

E6.11. Complete the following derivations by filling in justifications for each line.

a.
1. $A \leftrightarrow B$
2. $\quad A$
3. $\quad B$
4. $A \rightarrow B$

b.
1. $A \leftrightarrow B$
2. $\sim B$
3. $\quad A$
4. $\quad B$
5. $\quad \perp$
6. $\sim A$

c.
1. $\quad A \leftrightarrow \sim A$
2. $\quad\quad A$
3. $\quad\quad \sim A$
4. $\quad\quad \perp$
5. $\quad \sim A$
6. $\quad A$
7. $\quad \perp$
8. $\sim(A \leftrightarrow \sim A)$

d.
1. $\quad A$
2. $\quad\quad \sim A$
3. $\quad\quad A$
4. $\quad \sim A \rightarrow A$
5. $\quad \sim A \rightarrow A$
6. $\quad\quad \sim A$
7. $\quad\quad A$
8. $\quad\quad \perp$
9. $\quad A$
10. $A \leftrightarrow (\sim A \rightarrow A)$

e.

1.	$\sim A$	
2.	$\sim B$	
3.	$\quad A$	
4.	$\quad\quad \sim B$	
5.	$\quad\quad \bot$	
6.	$\quad B$	
7.	$\quad B$	
8.	$\quad\quad \sim A$	
9.	$\quad\quad \bot$	
10.	$\quad A$	
11.	$A \leftrightarrow B$	

E6.12. Each of the following are not legitimate *ND* derivations. In each case, explain why.

a.

1.	A	P
2.	B	P
3.	$A \leftrightarrow B$	1,2 $\leftrightarrow$I

b.

1.	$A \rightarrow B$	P
2.	B	P
3.	A	1,2 $\rightarrow$E

*c.

1.	$A \leftrightarrow B$	P
2.	A	1 $\leftrightarrow$E

d.

1.	B	P
2.	$\quad A$	A (g, $\leftrightarrow$I)
3.	$\quad B$	1 R
4.	$\quad B$	A (g, $\leftrightarrow$I)
5.	$\quad A$	2 R
6.	$A \leftrightarrow B$	2-3,4-5 $\leftrightarrow$I

e.

1.	$\sim A$	P
2.	$\quad \| \; B$	A (g, $\rightarrow$I)
3.	$\quad \| \; \| \; \sim A$	A (g, $\leftrightarrow$I)
4.	$\quad \| \; \| \; B$	2 R
5.	$\quad \| \; B$	2 R
6.	$B \rightarrow B$	2-5 $\rightarrow$I
7.	$\quad \| \; B$	A (g, $\leftrightarrow$I)
8.	$\quad \| \; \sim A$	1 R
9.	$\sim A \leftrightarrow B$	3-4,7-8 $\leftrightarrow$I

E6.13. Produce derivations to show each of the following.

*a. $(A \wedge B) \leftrightarrow A \vdash_{ND} A \rightarrow B$

b. $A \leftrightarrow (A \vee B) \vdash_{ND} B \rightarrow A$

c. $A \leftrightarrow B,\ B \leftrightarrow C,\ C \leftrightarrow D,\ \sim A \vdash_{ND} \sim D$

d. $A \leftrightarrow B \vdash_{ND} (A \rightarrow B) \wedge (B \rightarrow A)$

*e. $A \leftrightarrow (B \wedge C),\ B \vdash_{ND} A \leftrightarrow C$

f. $(A \rightarrow B) \wedge (B \rightarrow A) \vdash_{ND} (A \leftrightarrow B)$

g. $A \rightarrow (B \leftrightarrow C) \vdash_{ND} (A \wedge B) \leftrightarrow (A \wedge C)$

h. $A \leftrightarrow B,\ C \leftrightarrow D \vdash_{ND} (A \wedge C) \leftrightarrow (B \wedge D)$

i. $\vdash_{ND} A \leftrightarrow A$

j. $\vdash_{ND} (A \wedge B) \leftrightarrow (B \wedge A)$

*k. $\vdash_{ND} \sim\sim A \leftrightarrow A$

l. $\vdash_{ND} (A \leftrightarrow B) \rightarrow (B \leftrightarrow A)$

m. $(A \wedge B) \leftrightarrow (A \wedge C) \vdash_{ND} A \rightarrow (B \leftrightarrow C)$

n. $\sim A \rightarrow B,\ A \rightarrow \sim B \vdash_{ND} \sim A \leftrightarrow B$

o. $A,\ B \vdash_{ND} \sim A \leftrightarrow \sim B$

6.2.4 Strategies for a Goal

It is natural to introduce derivation rules, as we have, with relatively simple cases. And you may or may not have been able to see from the start in some cases how derivations would go. But derivations are not always so simple, and (short of genius) nobody can always see how they go. Perhaps this has already been an issue! So we want to think about derivation strategies. As we shall see later, for the quantificational case at least, it is not *possible* to produce a mechanical algorithm adequate to complete every completable derivation. However, as with chess or other games of strategy, it is possible to say a good deal about how to approach problems effectively. We have said quite a bit already. In this section, we pull together some of the themes, and present the material more systematically.

For natural derivation systems, the overriding strategy is to *work goal directedly*. What you do at any stage is directed primarily, not by what you have, but by where you want to be. Suppose you are trying to show that $\Gamma \vdash_{ND} \mathcal{P}$. You are given $\mathcal{P}$ as your goal. Perhaps it is tempting to begin by using E-rules to "see what you can get" from the members of Γ. There is nothing wrong with a bit of this in order to simplify your premises (like arranging the cards in your hand into some manageable order), but the main work of doing a derivation does not begin until you focus on the goal. This is not to say that your premises play no role in strategic thinking. Rather, it is to rule out doing things with them which are not purposefully directed at the end. In the ordinary case, applying the strategies for your goal dictates some new goal; applying strategies for this new goal dictates another; and so forth, until you come to a goal that is easily achieved.

The following *strategies for a goal* are arranged in rough priority order:

SG
1. If accessible lines contain explicit contradiction, use $\sim$E to reach goal.
2. Given an accessible formula with main operator $\vee$, use $\vee$E to reach goal.
3. If goal is "in" accessible lines (set goals and) attempt to exploit it out.
4. To reach goal with main operator $\star$, use $\star$I (careful with $\vee$).
5. Try $\sim$E (especially for atomics and sentences with $\vee$ as main operator).

If a high priority strategy applies, use it. If one does not apply, simply "fall through" to the next. The priority order is not necessarily a frequency order. The frequency will likely be something like SG4, SG3, SG5, SG2, SG1. But high priority strategies are such that you should adopt them if they are available — even though most often you will fall through to ones that are more frequently used. I take up the strategies in the priority order.

SG1 *If accessible lines contain explicit contradiction, use* ∼E *to reach goal*. For goal $\mathcal{B}$, with an explicit contradiction accessible, you can simply *assume* $\sim\mathcal{B}$, use your contradiction, and conclude $\mathcal{B}$.

given

$$\begin{array}{ll|ll} \text{a.} & & \mathcal{A} & \\ \text{b.} & & \sim\mathcal{A} & \\ & & & \\ & & \mathcal{B} & \text{(goal)} \end{array}$$

use

$$\begin{array}{ll|ll} \text{a.} & & \mathcal{A} & \\ \text{b.} & & \sim\mathcal{A} & \\ \text{c.} & & \quad \sim\mathcal{B} & \text{A } (c, \sim\text{E}) \\ \text{d.} & & \quad \bot & \text{a,b } \bot\text{I} \\ & & \mathcal{B} & \text{c-d } \sim\text{E} \end{array}$$

That is it! No matter what your goal is, given an accessible contradiction, you can reach that goal by ∼E. Since this strategy always delivers, you should jump on it whenever it is available. As an example, try to show, $A, \sim A \vdash_{ND} (R \wedge S) \rightarrow T$. Your derivation need not involve →I. Hint: I mean it! This section will be far more valuable if you work these examples, and so think through the steps. Here it is in two stages.

(AE)

$$\begin{array}{l|ll} 1. & A & \text{P} \\ 2. & \sim A & \text{P} \\ 3. & \quad \sim[(R \vee S) \rightarrow T] & \text{A } (c, \sim\text{E}) \\ & & \\ & (R \vee S) \rightarrow T & \end{array} \qquad \begin{array}{l|ll} 1. & A & \text{P} \\ 2. & \sim A & \text{P} \\ 3. & \quad \sim[(R \vee S) \rightarrow T] & \text{A } (c, \sim\text{E}) \\ 4. & \quad \bot & \text{1,2 } \bot\text{I} \\ 5. & (R \vee S) \rightarrow T & \text{3-4 } \sim\text{E} \end{array}$$

As soon as we see the accessible contradiction, we assume the negation of our goal, with a plan to exit by ∼E. This is accomplished on the left. Then it is a simple matter of applying the contradiction, and going to the conclusion by ∼E.

For this strategy, it is not required that accessible lines "contain" a contradiction only when it is directly available. However, the intent is that it should be no real work to obtain it. Perhaps an application of ∧E or the like does the job. It should be possible to obtain the contradiction immediately by some E-rule(s). If you can do this, then your derivation is over: assuming the opposite, applying the rules, and then ∼E reaches the goal. If there is no simple way to obtain a contradiction, fall through to the next strategy.

SG2 *Given an accessible formula with main operator* ∨, *use* ∨E *to reach goal*. As suggested above, you may prefer to avoid ∨E. But this is a mistake — ∨E is your friend! Suppose you have some accessible lines including a disjunction $\mathcal{A} \vee \mathcal{B}$ with goal $\mathcal{C}$. If you go for *that very goal* by ∨E, the result is a pair of subderivations with goal $\mathcal{C}$ — where, in the one case, all those very same accessible lines *and* $\mathcal{A}$ are

accessible, and in the other case, all those very same lines *and* $\mathscr{B}$ are accessible. So, in each subderivation, you can only be better off in your attempt to reach $\mathscr{C}$.

given

a.	$\mathscr{A} \vee \mathscr{B}$	
	C	(goal)

use

a.	$\mathscr{A} \vee \mathscr{B}$	
b.	$\quad\mathscr{A}$	A (g, a$\vee$E)
c.	$\quad\mathscr{C}$	(goal)
d.	$\quad\mathscr{B}$	A (g, a$\vee$E)
e.	$\quad\mathscr{C}$	(goal)
	$\mathscr{C}$	a,b-c,d-e $\vee$E

As an example, try to show, $A \rightarrow B,\ A \vee (A \wedge B) \vdash_{ND} A \wedge B$. Try showing it without $\vee$E! Here is the derivation in stages.

(AF)

1.	$A \rightarrow B$	P
2.	$A \vee (A \wedge B)$	P
3.	$\quad A$	A (g, 2$\vee$E)
	$\quad A \wedge B$	
	$\quad A \wedge B$	A (g, 2$\vee$E)
	$\quad A \wedge B$	
	$A \wedge B$	

1.	$A \rightarrow B$	P
2.	$A \vee (A \wedge B)$	P
3.	$\quad A$	A (g, 2$\vee$E)
4.	$\quad B$	1,3 $\rightarrow$E
5.	$\quad A \wedge B$	3,4 $\wedge$I
6.	$\quad A \wedge B$	A (g, 2$\vee$E)
7.	$\quad A \wedge B$	6 R
8.	$A \wedge B$	1,2-5,6-7 $\vee$E

When we start, there is no accessible contradiction. So we fall through to SG2. Since a premise has main operator $\vee$, we set up to get the goal by $\vee$E. This leads to a pair of simple subderivations. Once we do this, we treat the disjunction as effectively "used up" so that SG2 does not apply to it again. Notice that there is almost nothing one *could* do except set up this way — and that once you do, it is easy!

SG3 *If goal is "in" accessible lines (set goals and) attempt to exploit it out.* In most derivations, you will work toward goals which are successively closer to what can be obtained directly from accessible lines. And you finally come to a goal which can be obtained directly. If it can be obtained directly, do so! In some cases, however, you will come to a stage where your goal exists in accessible lines, but can be obtained only by means of some other result. In this case, you can set that other result as a *new* goal. A typical case is as follows.

given

a.	$\mathcal{A} \to \mathcal{B}$	
	$\mathcal{B}$	(goal)

use

a.	$\mathcal{A} \to \mathcal{B}$	
b.	$\mathcal{A}$	(goal)
	$\mathcal{B}$	a,b $\to$E

The $\mathcal{B}$ exists in the premises. You cannot get it without the $\mathcal{A}$. So you set $\mathcal{A}$ as a new goal and use it to get the $\mathcal{B}$. It is impossible to represent all the cases where this strategy applies. The idea is that the complete goal exists in accessible lines, and can either be obtained directly by an E-rule, or by an E-rule with some new goal. Observe that the strategy would not apply in case you have $A \to B$ and are going for A. Then the goal exists as part of a premise all right. But there is no obvious result such that obtaining it would give you a way to exploit $A \to B$ to get the A.

As an example, let us try to show $(A \to B) \wedge (B \to C)$, $(L \leftrightarrow S) \to A$, $(L \leftrightarrow S) \wedge H \vdash_{ND} C$. Here is the derivation in four stages.

(AG)

1.	$(A \to B) \wedge (B \to C)$	P
2.	$(L \leftrightarrow S) \to A$	P
3.	$(L \leftrightarrow S) \wedge H$	P
4.	$B \to C$	1 $\wedge$E
	B	
	C	4,_ $\to$E

1.	$(A \to B) \wedge (B \to C)$	P
2.	$(L \leftrightarrow S) \to A$	P
3.	$(L \leftrightarrow S) \wedge H$	P
4.	$B \to C$	1 $\wedge$E
5.	$A \to B$	1 $\wedge$E
	A	
	B	5,_ $\to$E
	C	4,_ $\to$E

The original goal C exists in the premises, as the consequent of the right conjunct of (1). It is easy to isolate the $B \to C$, but this leaves us with the B as a new goal to get the C. B also exists in the premises, as the consequent of the left conjunct of (1). Again, it is easy to isolate $A \to B$, but this leaves us with A as a new goal. We are not in a position to fill in the entire justification for our new goals, but there is no harm filling in what we can, to remind us where we are going. So far, so good.

1.	$(A \to B) \wedge (B \to C)$	P
2.	$(L \leftrightarrow S) \to A$	P
3.	$(L \leftrightarrow S) \wedge H$	P
4.	$B \to C$	1 $\wedge$E
5.	$A \to B$	1 $\wedge$E
	$L \leftrightarrow S$	
	A	2,_ $\to$E
	B	5,_ $\to$E
	C	4,_ $\to$E

1.	$(A \to B) \wedge (B \to C)$	P
2.	$(L \leftrightarrow S) \to A$	P
3.	$(L \leftrightarrow S) \wedge H$	P
4.	$B \to C$	1 $\wedge$E
5.	$A \to B$	1 $\wedge$E
6.	$L \leftrightarrow S$	3 $\wedge$E
7.	A	2,6 $\to$E
8.	B	5,7 $\to$E
9.	C	4,8 $\to$E

But A also exists in the premises, as the consequent of (2); to get it, we set $L \leftrightarrow S$ as a goal. But $L \leftrightarrow S$ exists in the premises, and is easy to get by $\wedge$E. So we complete the derivation with the steps that motivated the subgoals in the first place. Observe the way we move from one goal to the next, until finally there is a stage where SG3 applies in its simplest form, so that $L \leftrightarrow S$ is obtained directly.

SG4 *To reach goal with main operator $\star$, use $\star$I (careful with $\vee$).* This is the most frequently used strategy, the one most likely to structure your derivation as a whole. $\sim$E to the side, the basic structure of I-rules and E-rules in *ND* gives you just one way to generate a formula with main operator $\star$, whatever that may be. In the ordinary case, then, you can *expect* to obtain a formula with main operator $\star$ by the corresponding I-rule. Thus, for a typical example,

given

	$\mathcal{A} \rightarrow \mathcal{B}$	(goal)

use

a.	$\mathcal{A}$	A (g, $\rightarrow$I)
b.	$\mathcal{B}$	(goal)
	$\mathcal{A} \rightarrow \mathcal{B}$	a-b $\rightarrow$I

Again, it is difficult to represent all the cases where this strategy might apply. It makes sense to consider it for formulas with any main operator. Be cautious, however, for formulas with main operator $\vee$. There are cases where it is possible to prove a disjunction, but not to prove it by $\vee$I — as one might have conclusive reason to believe the butler *or* the maid did it, without conclusive reason to believe the butler did it, or conclusive reason to believe the maid did it (perhaps the butler and maid were the only ones with means and motive). You should consider the strategy for $\vee$. But it does not always work.

As an example, let us show $D \vdash_{ND} A \rightarrow (B \rightarrow (C \rightarrow D))$. Here is the derivation in four stages.

(AH)

1.	D	P
2.	A	A (g, $\rightarrow$I)
	$B \rightarrow (C \rightarrow D)$	
	$A \rightarrow (B \rightarrow (C \rightarrow D))$	2-_ $\rightarrow$I

1.	D	P
2.	A	A (g, $\rightarrow$I)
3.	B	A (g, $\rightarrow$I)
	$C \rightarrow D$	
	$B \rightarrow (C \rightarrow D)$	3-_ $\rightarrow$I
	$A \rightarrow (B \rightarrow (C \rightarrow D))$	2-_ $\rightarrow$I

Initially, there is no contradiction or disjunction in the premises, and neither do we see the goal. So we fall through to strategy SG4 and, since the main operator of the goal is $\rightarrow$, set up to get it by $\rightarrow$I. This gives us $B \rightarrow (C \rightarrow D)$ as a new goal. Since

this has main operator $\rightarrow$, and it remains that other strategies do not apply, we fall through to SG4, and set up to get it by $\rightarrow$I. This gives us $C \rightarrow D$ as a new goal.

1.	D	P	1.	D	P
2.	A	A (g, $\rightarrow$I)	2.	A	A (g, $\rightarrow$I)
3.	B	A (g, $\rightarrow$I)	3.	B	A (g, $\rightarrow$I)
4.	C	A $g, \rightarrow I)$	4.	C	A $g, \rightarrow I)$
	D		5.	D	1 R
	$C \rightarrow D$	4-_ $\rightarrow$I	6.	$C \rightarrow D$	4-5 $\rightarrow$I
	$B \rightarrow (C \rightarrow D)$	3-_ $\rightarrow$I	7.	$B \rightarrow (C \rightarrow D)$	3-6 $\rightarrow$I
	$A \rightarrow (B \rightarrow (C \rightarrow D))$	2-_ $\rightarrow$I	8.	$A \rightarrow (B \rightarrow (C \rightarrow D))$	2-7 $\rightarrow$I

As before, with $C \rightarrow D$ as the goal, there is no contradiction on accessible lines, no accessible formula has main operator $\vee$, and the goal does not itself appear on accessible lines. Since the main operator is $\rightarrow$, we set up again to get it by $\rightarrow$I. This gives us D as a new subgoal. But D does exist on an accessible line. Thus we are faced with a particularly simple instance of strategy SG3. To complete the derivation, we simply reiterate D from (1), and follow our exit strategies as planned.

SG5 *Try* $\sim$E (especially for atomics and sentences with $\vee$ as main operator). The previous strategy has no application to atomics, because they *have* no main operator, and we have suggested that it is problematic for disjunctions. This last strategy applies particularly in those cases. So it is applicable in cases where other strategies seem not to apply.

given

	$\mathcal{A}$	(goal)

use

a.	$\sim\mathcal{A}$	A (c, $\sim$E)
b.	$\bot$	
	$\mathcal{A}$	a-b $\sim$E

It is possible to obtain *any* formula by $\sim$E, by assuming the negation of it and going for a contradiction. So this strategy is generally applicable. And it cannot hurt: If you could have reached the goal anyway, you can obtain the goal $\mathcal{A}$ under the assumption, and then use *it* for a contradiction with the assumed $\sim\mathcal{A}$ — which lets you exit the assumption with the $\mathcal{A}$ you would have had anyway. And the assumption may help: for, as with $\vee$E, in going for the contradiction you have whatever accessible lines you had before, *plus* the new assumption. And, in many cases, the assumption puts you in a position to make progress you would not have been able to make before.

As a simple example of the strategy, try showing, $\sim A \rightarrow B, \sim B \vdash_{ND} A$. Here is the derivation in two stages.

(AI)

$$
\begin{array}{rlll}
1. & \sim A \rightarrow B & & \text{P} \\
2. & \sim B & & \text{P} \\
3. & \quad \sim A & & \text{A } (c, \sim\text{E}) \\
 & & & \\
 & \quad \bot & & \\
 & A & & 3\text{-}\_\ \sim\text{E}
\end{array}
\qquad\qquad
\begin{array}{rlll}
1. & \sim A \rightarrow B & & \text{P} \\
2. & \sim B & & \text{P} \\
3. & \quad \sim A & & \text{A } (c, \sim\text{E}) \\
4. & \quad B & & 1,3 \rightarrow\text{E} \\
5. & \quad \bot & & 4,2\ \bot\text{I} \\
6. & A & & 3\text{-}5 \sim\text{E}
\end{array}
$$

Sometimes the occasion between this strategy and SG1 can seem obscure (and, in the end, it may not be all that important to separate them). However, for the first, accessible lines *by themselves* are sufficient for a contradiction. In this example, from the premises we have $\sim B$, but cannot get the B and so do have a contradiction. So SG1 does not apply. There is no formula with main operator $\vee$. Similarly, though $\sim A$ is in the antecedent of (1), there is no obvious way to exploit the premise to isolate the A; so we do not see the goal in the relevant form in the premises. The goal A has no operators, so it has no main operator and strategy SG4 does not apply. So we fall through to strategy SG5, and set up to get the goal by $\sim$E. In this case, the subderivation is particularly easy to complete. Perhaps the case is too easy. Still, in contrast to SG1, the contradiction does not become available until after you make the assumption. In the case of SG1, it is the prior availability of the contradiction that drives your assumption.

Here is an extended example which combines a number of the strategies considered so far. We show that $B \vee A \vdash_{ND} \sim A \rightarrow B$. You want especially to absorb the *mode of thinking* about this case as a way to approach exercises.

(AJ)

$$
\begin{array}{rlll}
1. & B \vee A & & \text{P} \\
 & & & \\
 & \sim A \rightarrow B & &
\end{array}
$$

There is no contradiction in accessible premises; so strategy SG1 is inapplicable. Strategy SG2 tells us to go for the goal by $\vee$E. Another option is to fall through to SG4 and go for $\sim A \rightarrow B$ by $\rightarrow$I and then apply $\vee$E to get the B, but $\rightarrow$I has lower priority, and let us follow the official procedure.

1.	$B \vee A$	P	
2.	B	A (g, 1$\vee$E)	
	$\sim A \rightarrow B$		Given an accessible line with main operator $\vee$, use $\vee$E to reach goal.
	A	A (g, 1$\vee$E)	
	$\sim A \rightarrow B$		
	$\sim A \rightarrow B$	1, _, _ $\vee$E	

Having set up for $\vee$E on line (1), we treat $B \vee A$ as effectively "used up" and so out of the picture. Concentrating, for the moment, on the first subderivation, there is no contradiction on accessible lines; neither is there another accessible disjunction; and the goal is not in the premises. So we fall through to SG4.

1.	$B \vee A$	P	
2.	B	A (g, 1$\vee$E)	
3.	$\sim A$	A (g, $\rightarrow$I)	
	B		
	$\sim A \rightarrow B$	3-_ $\rightarrow$I	To reach goal with main operator $\rightarrow$, use $\rightarrow$I.
	A	A (g, 1$\vee$E)	
	$\sim A \rightarrow B$		
	$\sim A \rightarrow B$	1, _, _ $\vee$E	

In this case, the subderivation is easy to complete. The new goal, B exists as such in the premises. So we are faced with a simple instance of SG3, and so can complete the subderivation.

1.	$B \vee A$	P	
2.	B	A (g, 1$\vee$E)	
3.	$\sim A$	A (g, $\rightarrow$I)	
4.	B	2 R	The first subderivation is completed by reiterating B from line (2), and following the exit strategy.
5.	$\sim A \rightarrow B$	3-4 $\rightarrow$I	
6.	A	A (g, 1$\vee$E)	
	$\sim A \rightarrow B$		
	$\sim A \rightarrow B$	1, _, _ $\vee$E	

For the second main subderivation tick off in your head: there is no accessible contradiction; neither is there another accessible formula with main operator $\vee$; and the goal is not in the premises. So we fall through to strategy SG4.

1.	$\mid B \vee A$	P
2.	$\mid \; \mid B$	A (g, 1$\vee$E)
3.	$\mid \; \mid \; \mid {\sim}A$	A (g, $\rightarrow$I)
4.	$\mid \; \mid \; \mid B$	2 R
5.	$\mid \; \mid {\sim}A \rightarrow B$	3-4 $\rightarrow$I
6.	$\mid \; \mid A$	A (g, 1$\vee$E)
7.	$\mid \; \mid \; \mid {\sim}A$	A (g, $\rightarrow$I)
	$\mid \; \mid \; \mid$	
	$\mid \; \mid \; \mid B$	
	$\mid \; \mid {\sim}A \rightarrow B$	7-_ $\rightarrow$I
	$\mid {\sim}A \rightarrow B$	1,_,_ $\vee$E

To reach goal with main operator $\rightarrow$, use $\rightarrow$I.

In this case, there *is* an accessible contradiction at (6) and (7). So SG1 applies, and we are in a position to complete the derivation as follows.

1.	$\mid B \vee A$	P
2.	$\mid \; \mid B$	A (g, 1$\vee$E)
3.	$\mid \; \mid \; \mid {\sim}A$	A (g, $\rightarrow$I)
4.	$\mid \; \mid \; \mid B$	2 R
5.	$\mid \; \mid {\sim}A \rightarrow B$	3-4 $\rightarrow$I
6.	$\mid \; \mid A$	A (g, 1$\vee$E)
7.	$\mid \; \mid \; \mid {\sim}A$	A (g, $\rightarrow$I)
8.	$\mid \; \mid \; \mid \; \mid {\sim}B$	A (c, $\sim$E)
9.	$\mid \; \mid \; \mid \; \mid \bot$	6,7 $\bot$I
10.	$\mid \; \mid \; \mid B$	8-9 $\sim$E
11.	$\mid \; \mid {\sim}A \rightarrow B$	7-10 $\rightarrow$I
12.	$\mid {\sim}A \rightarrow B$	1,2-5,6-11 $\vee$E

If accessible lines contain explicit contradiction, use $\sim$E to reach goal.

This derivation is fairly complicated! But we did not need to see how the whole thing would go from the start. Indeed, it is hard to see how one could do so. Rather it was enough to see, at each stage, what to do next. That is the beauty of our goal-oriented approach.

A couple of final remarks before we turn to exercises: First, as we have said from the start, assumptions are only introduced in conjunction with exit strategies. This

almost requires goal-directed thinking. And it is important to see how pointless are assumptions without an exit strategy! Results inside subderivations cannot be used for a final conclusion except insofar as there is a way to exit the subderivation and use it whole. So the point of the strategy is to ensure that the subderivation has a use for getting where you want to go.

Second, in going for a contradiction, as with SG4 or SG5, the new goal is not a definite formula — any contradiction is sufficient for the rule and for a derivation of $\perp$. So the strategies for a goal do not directly apply. This motivates the "strategies for a contradiction" of the next section. For now, I will say just this: If there is a contradiction to be had, and you can reduce formulas on accessible lines to atomics and negated atomics, the contradiction *will* appear at that level. So one way to go for a contradiction is simply by applying E-rules to accessible lines, to generate what atomics and negated atomics you can.

Proof for the following theorems are left as exercises. You should not start them now, but wait for the assignment in E6.16. The first three may remind you of axioms from chapter 3. The others foreshadow rules from the system *ND+*, which we will see shortly.

T6.1. $\vdash_{ND} \mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$

T6.2. $\vdash_{ND} (\mathcal{O} \to (\mathcal{P} \to \mathcal{Q})) \to ((\mathcal{O} \to \mathcal{P}) \to (\mathcal{O} \to \mathcal{Q}))$

*T6.3. $\vdash_{ND} (\sim\mathcal{Q} \to \sim\mathcal{P}) \to ((\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q})$

T6.4. $\mathcal{A} \to \mathcal{B}, \sim\mathcal{B} \vdash_{ND} \sim\mathcal{A}$

T6.5. $\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C} \vdash_{ND} \mathcal{A} \to \mathcal{C}$

T6.6. $\mathcal{A} \vee \mathcal{B}, \sim\mathcal{A} \vdash_{ND} \mathcal{B}$

T6.7. $\mathcal{A} \vee \mathcal{B}, \sim\mathcal{B} \vdash_{ND} \mathcal{A}$

T6.8. $\mathcal{A} \leftrightarrow \mathcal{B}, \sim\mathcal{A} \vdash_{ND} \sim\mathcal{B}$

T6.9. $\mathcal{A} \leftrightarrow \mathcal{B}, \sim\mathcal{B} \vdash_{ND} \sim\mathcal{A}$

T6.10. $\vdash_{ND} (\mathcal{A} \wedge \mathcal{B}) \leftrightarrow (\mathcal{B} \wedge \mathcal{A})$

*T6.11. $\vdash_{ND} (\mathcal{A} \vee \mathcal{B}) \leftrightarrow (\mathcal{B} \vee \mathcal{A})$

T6.12. $\vdash_{ND} (\mathcal{A} \rightarrow \mathcal{B}) \leftrightarrow (\sim\mathcal{B} \rightarrow \sim\mathcal{A})$

T6.13. $\vdash_{ND} [\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})] \leftrightarrow [(\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{C}]$

T6.14. $\vdash_{ND} [\mathcal{A} \wedge (\mathcal{B} \wedge \mathcal{C})] \leftrightarrow [(\mathcal{A} \wedge \mathcal{B}) \wedge \mathcal{C}]$

T6.15. $\vdash_{ND} [\mathcal{A} \vee (\mathcal{B} \vee \mathcal{C})] \leftrightarrow [(\mathcal{A} \vee \mathcal{B}) \vee \mathcal{C}]$

T6.16. $\vdash_{ND} \mathcal{A} \leftrightarrow \sim\sim\mathcal{A}$

T6.17. $\vdash_{ND} \mathcal{A} \leftrightarrow (\mathcal{A} \wedge \mathcal{A})$

T6.18. $\vdash_{ND} \mathcal{A} \leftrightarrow (\mathcal{A} \vee \mathcal{A})$

E6.14. For each of the following, (i) which goal strategy applies? and (ii) what is the next step? If the strategy calls for a new subgoal, show the subgoal; if it calls for a subderivation, set up the subderivation. In each case, *explain* your response. Hint: Each goal strategy applies once.

a.
$$\begin{array}{r|ll} 1. & \sim A \vee B & \text{P} \\ 2. & A & \text{P} \\ \hline & & \\ & B & \end{array}$$

b.
$$\begin{array}{r|ll} 1. & J \wedge S & \text{P} \\ 2. & S \rightarrow K & \text{P} \\ \hline & & \\ & K & \end{array}$$

*c.

1.	$\sim A \leftrightarrow B$	P
	$B \leftrightarrow \sim A$	

d.

1.	$A \leftrightarrow \sim B$	P
2.	$\sim A$	P
	B	

e.

1.	$A \wedge B$	P
2.	$\sim A$	P
	$K \vee J$	

E6.15. Produce derivations to show each of the following. No worked out answers are provided. However, if you get stuck, you will find strategy hints in the back.

*a. $A \leftrightarrow (A \rightarrow B) \vdash_{ND} A \rightarrow B$

*b. $(A \vee B) \rightarrow (B \leftrightarrow D), B \vdash_{ND} B \wedge D$

*c. $\sim(A \wedge C), \sim(A \wedge C) \leftrightarrow B \vdash_{ND} A \vee B$

*d. $A \wedge (C \wedge \sim B), (A \vee D) \rightarrow \sim E \vdash_{ND} \sim E$

*e. $A \rightarrow B, B \rightarrow C \vdash_{ND} A \rightarrow C$

*f. $(A \wedge B) \rightarrow (C \wedge D) \vdash_{ND} [(A \wedge B) \rightarrow C] \wedge [(A \wedge B) \rightarrow D]$

*g. $A \rightarrow (B \rightarrow C), (A \wedge D) \rightarrow E, C \rightarrow D \vdash_{ND} (A \wedge B) \rightarrow E$

*h. $(A \rightarrow B) \wedge (B \rightarrow C), [(D \vee E) \vee H] \rightarrow A, \sim(D \vee E) \wedge H \vdash_{ND} C$

*i. $A \rightarrow (B \wedge C), \sim C \vdash_{ND} \sim(A \wedge D)$

*j. $A \rightarrow (B \rightarrow C), D \rightarrow B \vdash_{ND} A \rightarrow (D \rightarrow C)$

*k. $A \rightarrow (B \rightarrow C) \vdash_{ND} \sim C \rightarrow \sim(A \wedge B)$

*l. $(A \wedge \sim B) \rightarrow \sim A \vdash_{ND} A \rightarrow B$

*m. $\sim B \leftrightarrow A, C \rightarrow B, A \wedge C \vdash_{ND} \sim K$

*n. $\sim A \vdash_{ND} A \rightarrow B$

*o. $\sim A \leftrightarrow \sim B \vdash_{ND} A \leftrightarrow B$

*p. $(A \vee B) \vee C, B \leftrightarrow C \vdash_{ND} C \vee A$

*q. $\vdash_{ND} A \rightarrow (A \vee B)$

*r. $\vdash_{ND} A \rightarrow (B \rightarrow A)$

*s. $\vdash_{ND} (A \leftrightarrow B) \rightarrow (A \rightarrow B)$

*t. $\vdash_{ND} (A \wedge \sim A) \rightarrow (B \wedge \sim B)$

*u. $\vdash_{ND} (A \rightarrow B) \rightarrow [(C \rightarrow A) \rightarrow (C \rightarrow B)]$

*v. $\vdash_{ND} [(A \rightarrow B) \wedge \sim B] \rightarrow \sim A$

*w. $\vdash_{ND} A \rightarrow [B \rightarrow (A \rightarrow B)]$

*x. $\vdash_{ND} \sim A \rightarrow [(B \wedge A) \rightarrow C]$

*y. $\vdash_{ND} (A \rightarrow B) \rightarrow [\sim B \rightarrow \sim(A \wedge D)]$

*E6.16. Produce derivations to demonstrate each of T6.1 - T6.18. This is a mix — some repetitious, some challenging! But, when we need the results later, we will be glad to have done them now. Hint: do not worry if one or two get a bit longer than you are used to — they should!

6.2.5 Strategies for a Contradiction

In going for a contradiction, the $\mathcal{Q}$ and $\sim\mathcal{Q}$ can be any sentence. So the strategies for reaching a definite goal do not apply. This motivates *strategies for a contradiction*. Again, the strategies are in rough priority order.

SC

1. Break accessible formulas down into atomics and negated atomics.
2. Given a disjunction in a subderivation for $\sim$E or $\sim$I, go for $\perp$ by $\vee$E.
3. Set as goal the opposite of some negation (something that cannot itself be broken down). Then apply strategies for a goal to reach it.
4. For some $\mathcal{P}$ such that both $\mathcal{P}$ and $\sim\mathcal{P}$ lead to contradiction: Assume $\mathcal{P}$ ($\sim\mathcal{P}$), obtain the first contradiction, and conclude $\sim\mathcal{P}$ ($\mathcal{P}$); then obtain the second contradiction — this is the one you want.

Again, the priority order is not the frequency order. The frequency is likely to be something like SC1, SC3, SC4, SC2. Also sometimes, but not always, SC3 and SC4 coincide: in deriving the opposite of some negation, you end up assuming a $\mathcal{P}$ such that $\mathcal{P}$ and $\sim\mathcal{P}$ lead to contradiction.

SC1. *Break accessible formulas down into atomics and negated atomics.* As we have already said, if there is a contradiction to be had, and you can break premises into atomics and negated atomics, the contradiction *will* appear at that level. Thus, for example,

(AK)

1.	$A \wedge B$	P
2.	$\sim B$	P
3.	$\quad C$	A (c, $\sim$I)
	$\quad \bot$	
	$\sim C$	2-__ $\sim$I

1.	$A \wedge B$	P
2.	$\sim B$	P
3.	$\quad C$	A (c, $\sim$I)
4.	$\quad A$	1 $\wedge$E
5.	$\quad B$	1 $\wedge$E
6.	$\quad \bot$	5,2 $\bot$I
7.	$\sim C$	2-6 $\sim$I

Our strategy for the goal, is SG4 with an application of $\sim$I. Then the goal is to obtain a contradiction. And our first thought is to break accessible lines down to atomics and negated atomics. Perhaps this example is too simple. And you may wonder about the point of getting A at (4) — there *is* no need for A at (4). But this merely illustrates the point: if you can get to atomics and negated atomics ("randomly" as it were) the contradiction will appear in the end.

As another example, try showing $A \wedge (B \wedge \sim C), \sim F \rightarrow D, (A \wedge D) \rightarrow C \vdash_{ND} F$. Here is the completed derivation in two stages.

(AL)

1.	$A \wedge (B \wedge \sim C)$	P
2.	$\sim F \rightarrow D$	P
3.	$(A \wedge D) \rightarrow C$	P
4.	$\quad \sim F$	A (c, $\sim$E)
	$\quad \bot$	
	F	4-__ $\sim$E

1.	$A \wedge (B \wedge \sim C)$	P
2.	$\sim F \rightarrow D$	P
3.	$(A \wedge D) \rightarrow C$	P
4.	$\quad \sim F$	A (c, $\sim$E)
5.	$\quad D$	2,4 $\rightarrow$E
6.	$\quad A$	1 $\wedge$E
7.	$\quad A \wedge D$	6,5 $\wedge$I
8.	$\quad C$	3,7 $\rightarrow$E
9.	$\quad B \wedge \sim C$	1 $\wedge$E
10.	$\quad \sim C$	9 $\wedge$E
11.	$\quad \bot$	8,10 $\bot$I
11.	F	4-10 $\sim$E

This time, our strategy for the goal, falls through to SG5. After that, again, our goal is to obtain a contradiction — and our first thought is to break premises down to atomics and negated atomics. The assumption $\sim F$ gets us D with (2). We can get A from (1), and then C with the A and D together. Then $\sim C$ follows from (1) by a couple applications of $\wedge$E. You might proceed to get the atomics in a different order, but the basic idea of any such derivation is likely to be the same.

SC2. *Given a disjunction in a subderivation for* $\sim$*E or* $\sim$*I, go for* $\perp$ *by* $\vee$*E.* This strategy applies only occasionally, though it is related to one that is common for the quantificational case. In most cases, you will have applied $\vee$E by SG2 prior to setting up for $\sim$E or $\sim$I. In some cases, however, a disjunction is "uncovered" only inside a subderivation for a tilde rule. In any such case, SC2 has high priority for the same reasons as SG2: You can only be *better off* in your attempt to reach a contradiction inside the subderivations for $\vee$E than before. So the strategy says to set $\perp$ as the goal you need for $\sim$E or $\sim$I, and go for *it* by $\vee$E.

given

a.	$\mathcal{P}$	A (c, $\sim$I)
b.	$\mathcal{A} \vee \mathcal{B}$	
	$\perp$	
	$\sim\mathcal{P}$	a-_ $\sim$I

use

a.	$\mathcal{P}$	A (c, $\sim$I)
b.	$\mathcal{A} \vee \mathcal{B}$	
c.	$\mathcal{A}$	A (c, b$\vee$E)
d.	$\perp$	
e.	$\mathcal{B}$	A (c, c$\vee$E)
f.	$\perp$	
g.	$\perp$	b,c-d,e-f $\vee$E
	$\sim\mathcal{P}$	a-g $\sim$I

Observe that, since the subderivations for $\vee$E have goal $\perp$, they have exit strategy c rather than g. Here is another advantage of our standard use of $\perp$. Because $\perp$ is a particular sentence, it works as a goal sentence for this rule. We might obtain $\perp$ by one contradiction in the first subderivation, and by another in the second. But, once we have obtained $\perp$ in each, we are in a position to exit by $\vee$E in the usual way, and so to apply $\sim$I.

Here is an example. We show $\sim A \wedge \sim B \vdash_{ND} \sim(A \vee B)$. The derivation is in four stages.

(AM)

1.	$\sim A \wedge \sim B$	P
2.	$A \vee B$	A (c, $\sim$I)
	$\perp$	
	$\sim(A \vee B)$	2-_ $\sim$I

1.	$\sim A \wedge \sim B$	P
2.	$A \vee B$	A (c, $\sim$I)
3.	A	A (c, 2$\vee$E)
	$\perp$	
	B	A (c, 2$\vee$E)
	$\perp$	
	$\perp$	2,_,_ $\vee$E
	$\sim(A \vee B)$	2-_ $\sim$I

In this case, our strategy for the goal is SG4. The disjunction appears only inside the subderivation as the assumption for $\sim$I. We might obtain $\sim A$ and $\sim B$ from (1) but after that, there are no more atomics or negated atomics to be had. So we fall through to SC2, with $\perp$ as the goal for $\vee$E.

1.	$\sim A \wedge \sim B$	P
2.	$A \vee B$	A (c, $\sim$I)
3.	A	A (c, 2$\vee$E)
4.	$\sim A$	1 $\wedge$E
5.	$\perp$	3,4 $\perp$I
6.	B	A (c, 2$\vee$E)
	$\perp$	
	$\perp$	2,3-5,_ $\vee$E
	$\sim(A \vee B)$	2-_ $\sim$I

1.	$\sim A \wedge \sim B$	P
2.	$A \vee B$	A (c, $\sim$I)
3.	A	A (c, 2$\vee$E)
4.	$\sim A$	1 $\wedge$E
5.	$\perp$	3,4 $\perp$I
6.	B	A (c, 2$\vee$E)
7.	$\sim B$	1 $\wedge$E
8.	$\perp$	6,7 $\perp$I
9.	$\perp$	2,3-5,6-8 $\vee$E
10.	$\sim(A \vee B)$	2-11 $\sim$I

The first subderivation is easily completed from atomics and negated atomics. And the second is completed the same way. Observe that it is only because of our assumptions for $\vee$E that we are able to get the contradictions at all.

SC3. *Set as goal the opposite of some negation (something that cannot itself be broken down). Then apply standard strategies for the goal.* You will find yourself using this strategy often, after SC1. In the ordinary case, if accessible formulas cannot be broken into atomics and negated atomics, it is because complex forms are "sealed off" by main operator $\sim$. The tilde blocks SC1 or SC2. But you can turn this lemon to lemonade: taking the complex $\sim\mathcal{Q}$ as one half of a contradiction, set $\mathcal{Q}$ as goal. For some complex $\mathcal{Q}$,

given

a.	$\sim\mathcal{Q}$	
b.	$\quad \mathcal{A}$	A (c, $\sim$I)
	$\quad \bot$	
	$\sim\mathcal{A}$	

use

a.	$\sim\mathcal{Q}$	
b.	$\quad \mathcal{A}$	A (c, $\sim$I)
c	$\quad \mathcal{Q}$	(goal)
	$\quad \bot$	c,a $\bot$I
	$\sim\mathcal{A}$	

We are after a contradiction. Supposing that we cannot break $\sim\mathcal{Q}$ into its parts, our efforts to apply other strategies for a contradiction are frustrated. But SC3 offers an alternative: Set $\mathcal{Q}$ itself as a new goal and use this with $\sim\mathcal{Q}$ to reach $\bot$. Then strategies for the new goal take over. If we reach the new goal, we have the contradiction we need.

As an example, try showing $B, \sim(A \rightarrow B) \vdash_{ND} \sim A$. Here is the derivation in four stages.

(AN)

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\quad A$	A (c, $\sim$I)
	$\quad \bot$	
	$\sim A$	3-__ $\sim$I

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\quad A$	A (c, $\sim$I)
	$\quad A \rightarrow B$	(goal)
	$\quad \bot$	__,2 $\bot$I
	$\sim A$	3-__ $\sim$I

Our strategy for the goal is SG4; for main operator $\sim$ we set up to get the goal by $\sim$I. So we need a contradiction. In this case, there is nothing to be done by way of obtaining atomics and negated atomics, and there is no disjunction in the scope of the assumption for $\sim$I. So we fall through to strategy SC3. $\sim(A \rightarrow B)$ on (2) has main operator $\sim$, so we set $A \rightarrow B$ as a new subgoal with the idea to use it for contradiction.

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\quad A$	A (c, $\sim$I)
4.	$\quad\quad A$	A (g, $\rightarrow$I)
	$\quad\quad B$	(goal)
	$\quad A \rightarrow B$	4-__ $\rightarrow$I
	$\quad \bot$	__,2 $\bot$I
	$\sim A$	3-__ $\sim$I

1.	B	P
2.	$\sim(A \rightarrow B)$	P
3.	$\quad A$	A (c, $\sim$I)
4.	$\quad\quad A$	A (g, $\rightarrow$I)
5.	$\quad\quad B$	1 R
6.	$\quad A \rightarrow B$	4-5 $\rightarrow$I
7.	$\quad \bot$	6,2 $\bot$I
8.	$\sim A$	3-7 $\sim$I

Since $A \rightarrow B$ is a definite subgoal, we proceed with strategies for the goal in the usual way. The main operator is $\rightarrow$ so we set up to get it by $\rightarrow$I. The subderivation

is particularly easy to complete. And we finish by executing the exit strategies as planned.

SC4. *For some $\mathscr{P}$ such that both $\mathscr{P}$ and $\sim\mathscr{P}$ lead to contradiction*: Assume $\mathscr{P}$ ($\sim\mathscr{P}$), obtain the first contradiction, and conclude $\sim\mathscr{P}$ ($\mathscr{P}$); then obtain the second contradiction — this is the one you want.

given

$$\begin{array}{ll|l|ll}
\text{a.} & & \mathscr{A} & & \text{A } (c, \sim\text{I}) \\
 & & & & \\
 & & \bot & & \\
 & \multicolumn{2}{l}{\sim\mathscr{A}} & &
\end{array}$$

use

$$\begin{array}{lll}
\text{a.} & \quad \mathscr{A} & \text{A } (c, \sim\text{I}) \\
\text{b.} & \qquad \mathscr{P} & \text{A } (c, \sim\text{I}) \\
\text{c.} & \qquad \bot & \\
 & \quad \sim\mathscr{P} & \text{b-c } \sim\text{I} \\
\text{d.} & \quad \bot & \\
 & \sim\mathscr{A} & \text{a-d } \sim\text{I}
\end{array}$$

The essential point is that both $\mathscr{P}$ and $\sim\mathscr{P}$ somehow lead to contradiction. Thus the assumption of one leads by $\sim$I or $\sim$E to the other; and since *both* lead to contradiction, you end up with the contradiction you need. This is often a powerful way of making progress when none seems possible by other means.

Let us try to show $A \leftrightarrow B$, $B \leftrightarrow C$, $C \leftrightarrow \sim A \vdash_{ND} K$. Here is the derivation in four stages.

(AO)

$$\begin{array}{lll}
1. & A \leftrightarrow B & \text{P} \\
2. & B \leftrightarrow C & \text{P} \\
3. & C \leftrightarrow \sim A & \text{P} \\
4. & \quad \sim K & \text{A } (c, \sim\text{E}) \\
 & \quad \bot & \\
 & K & \text{4-_ } \sim\text{E}
\end{array}$$

$$\begin{array}{lll}
1. & A \leftrightarrow B & \text{P} \\
2. & B \leftrightarrow C & \text{P} \\
3. & C \leftrightarrow \sim A & \text{P} \\
4. & \quad \sim K & \text{A } (c, \sim\text{E}) \\
5. & \qquad A & \text{A } (c, \sim\text{I}) \\
 & \qquad \bot & \\
 & \quad \sim A & \text{5-_ } \sim\text{I} \\
 & \quad \bot & \\
 & K & \text{4-_ } \sim\text{E}
\end{array}$$

Our strategy for the goal falls all the way through to SG5. So we assume the negation of the goal, and go for a contradiction. In this case, there are no atomics or negated atomics to be had. There is no disjunction under the scope of the negation, and no formula is itself a negation such that we could reiterate and build up to the opposite.

But given formula A we can use $\leftrightarrow$E to reach $\sim A$ and so contradiction. And, similarly, given $\sim A$ we can use $\leftrightarrow$E to reach A and so contradiction. So, following SC4, we assume one of them to get the other.

1.	$A \leftrightarrow B$	P
2.	$B \leftrightarrow C$	P
3.	$C \leftrightarrow \sim A$	P
4.	$\sim K$	A (c, $\sim$E)
5.	A	A (c, $\sim$I)
6.	B	1,5 $\leftrightarrow$E
7.	C	2,6 $\leftrightarrow$E
8.	$\sim A$	3,7 $\leftrightarrow$E
9.	$\bot$	5,8 $\bot$I
10.	$\sim A$	5-9 $\sim$I
	$\bot$	
	K	4-_ $\sim$E

1.	$A \leftrightarrow B$	P
2.	$B \leftrightarrow C$	P
3.	$C \leftrightarrow \sim A$	P
4.	$\sim K$	A (c, $\sim$E)
5.	A	A (c, $\sim$I)
6.	B	1,5 $\leftrightarrow$E
7.	C	2,6 $\leftrightarrow$E
8.	$\sim A$	3,7 $\leftrightarrow$E
9.	$\bot$	5,8 $\bot$I
10.	$\sim A$	5-9 $\sim$I
11.	C	3,10 $\leftrightarrow$E
12.	B	2,11 $\leftrightarrow$E
13.	A	1,12 $\leftrightarrow$E
14.	$\bot$	13,10 $\bot$I
15.	K	4-14 $\sim$E

The first contradiction appears easily at the level of atomics and negated atomics. This gives us $\sim A$. And with $\sim A$, the second contradiction also comes easily, at the level of atomics and negated atomics.

Though it can be useful, this strategy is often difficult to see. And there is no obvious way to give a strategy for using the strategy! The best thing to say is that you should *look for it* when the other strategies seem to fail.

Let us consider an extended example which combines some of the strategies. We show that $\sim A \to B \vdash_{ND} B \vee A$.

(AP)

1.	$\sim A \to B$	P
	$B \vee A$	

In this case, we do not see a contradiction in the premises; there is no formula with main operator $\vee$ in the premises; and the goal does not appear in the premises. So we might try going for the goal by $\vee$I in application of SG4. This would require getting a B or an A. It is reasonable to go this way, but it turns out to be a dead end. (You should convince yourself that this is so.) Thus we fall through to SG5.

$$
\begin{array}{rlll}
1. & \sim A \to B & & \text{P} \\
2. & \quad \sim(B \vee A) & & \text{A } (c, \sim\text{E}) \\
 & & & \\
 & \quad \bot & & \\
 & B \vee A & & 2\text{-}\_\ \sim\text{E}
\end{array}
$$

Especially considering our goal has main operator $\vee$, set up to get the goal by $\sim$E.

To get a contradiction, our first thought is to go for atomics and negated atomics. But there is nothing to be done. Similarly, there is no formula with main operator $\vee$. So we fall through to SC3 and continue as follows.

$$
\begin{array}{rlll}
1. & \sim A \to B & & \text{P} \\
2. & \quad \sim(B \vee A) & & \text{A } (c, \sim\text{E}) \\
 & & & \\
 & \quad B \vee A & & \\
 & \quad \bot & & \_,2\ \bot\text{I} \\
 & B \vee A & & 2\text{-}\_\ \sim\text{E}
\end{array}
$$

Given a negation that cannot be broken down, set up to get the contradiction by building up to the opposite.

It might seem that we have made no progress, since our new goal is no different than the original! But there is progress insofar as we have a premise not available before (more on this in a moment). At this stage, we *can* get the goal by $\vee$I. Either side will work, but it is easier to start with the A. So we set up for that.

$$
\begin{array}{rlll}
1. & \sim A \to B & & \text{P} \\
2. & \quad \sim(B \vee A) & & \text{A } (c, \sim\text{E}) \\
 & & & \\
 & \quad A & & \\
 & \quad B \vee A & & \_\ \vee\text{I} \\
 & \quad \bot & & \_,2\ \bot\text{I} \\
 & B \vee A & & 2\text{-}\_\ \sim\text{E}
\end{array}
$$

For a goal with main operator $\vee$, go for the goal by $\vee$I

Now the goal is atomic. Again, there is no contradiction or formula with main operator $\vee$ in the premises. The goal is not in the premises in any form we can hope to exploit. And the goal has no main operator. So, again, we fall through to SG5.

1.	$\sim A \to B$	P	
2.	$\vert\ \sim(B \vee A)$	A (c, $\sim$E)	
3.	$\vert\ \vert\ \sim A$	A (c, $\sim$E)	
	$\vert\ \vert\ \vert$		Especially for atomics, go for the goal by $\sim$E
	$\vert\ \vert\ \vert\ \bot$		
	$\vert\ \vert\ A$	3-_ $\sim$E	
	$\vert\ \vert\ B \vee A$	_ $\vee$I	
	$\vert\ \vert\ \bot$	_,2 $\bot$I	
	$\vert\ B \vee A$	2-_ $\sim$E	

Again, our first thought is to get atomics and negated atomics. We can get B from lines (1) and (3) by $\to$E. But that is all. So we will not get a contradiction from atomics and negated atomics alone. There is no formula with main operator $\vee$. However, the possibility of getting a B suggests that we *can* build up to the opposite of line (2). That is, we complete the subderivation as follows, and follow our exit strategies to complete the whole.

1.	$\sim A \to B$	P	
2.	$\vert\ \sim(B \vee A)$	A (c, $\sim$E)	
3.	$\vert\ \vert\ \sim A$	A (c, $\sim$E)	
4.	$\vert\ \vert\ \vert\ B$	1,3 $\to$E	Get the contradiction by building up to the opposite of an existing negation.
5.	$\vert\ \vert\ \vert\ B \vee A$	4 $\vee$I	
6.	$\vert\ \vert\ \vert\ \bot$	5,2 $\bot$I	
7.	$\vert\ \vert\ A$	3-6 $\sim$E	
8.	$\vert\ \vert\ B \vee A$	7 $\vee$I	
9.	$\vert\ \vert\ \bot$	8,2 $\bot$I	
10.	$\vert\ B \vee A$	2-9 $\sim$E	

A couple of comments: First, observe that we build up to the opposite of $\sim(B \vee A)$ *twice*, coming at it from different directions. First we obtain the left side B and use $\vee$I to obtain the whole, then the right side A and use $\vee$I to obtain the whole. This is typical with negated disjunctions. Second, note that this derivation might be reconceived as an instance of SC4. $\sim A$ gets us B, and so $B \vee A$, which contradicts $\sim(B \vee A)$. But A gets us $B \vee A$ which, again, contradicts $\sim(B \vee A)$. So both A and $\sim A$ lead to contradiction; so we assume one ($\sim A$), and get the first contradiction; this gets us A, from which the second contradiction follows.

The general pattern of this derivation is typical for formulas with main operator $\vee$ in *ND*. For $\mathcal{P} \vee \mathcal{Q}$ we may not be able to prove either $\mathcal{P}$ or $\mathcal{Q}$ from scratch — so that the formula is not directly provable by $\vee$I. However, it may be *indirectly* provable. If it is provable at all, it *must* be that the negation of one side forces the other. So it

must be possible to get the $\mathscr{P}$ or the $\mathscr{Q}$ under the *additional* assumption that the other is false. This makes possible an argument of the following form.

$$
\text{(AQ)}\qquad
\begin{array}{rlll}
\text{a.} & \mid\ \mid & \sim(\mathscr{P} \vee \mathscr{Q}) & \text{A } (c, \sim\text{E} \\
\text{b.} & \mid\ \mid\ \mid & \sim\mathscr{P} & \text{A } (c, \sim\text{E}) \\
 & \mid\ \mid\ \mid & \vdots & \\
\text{c.} & \mid\ \mid\ \mid & \mathscr{Q} & \\
\text{d.} & \mid\ \mid\ \mid & \mathscr{P} \vee \mathscr{Q} & \text{c } \vee\text{I} \\
\text{e.} & \mid\ \mid\ \mid & \bot & \text{d,a } \bot\text{I} \\
\text{f.} & \mid\ \mid & \mathscr{P} & \text{b-e } \sim\text{E} \\
\text{g.} & \mid\ \mid & \mathscr{P} \vee \mathscr{Q} & \text{f } \vee\text{I} \\
\text{h.} & \mid\ \mid & \bot & \text{g,a } \bot\text{I} \\
\text{i.} & \mid & \mathscr{P} \vee \mathscr{Q} & \text{a-h } \sim\text{E}
\end{array}
$$

The "work" in this routine is getting from the negation of one side of the disjunction to the other. Thus if from the assumption $\sim\mathscr{P}$ it is possible to derive $\mathscr{Q}$, all the rest is automatic! We have just seen an extended example (AP) of this pattern. It may be seen as an application of SC3 or SC4 (or both). Where a disjunction may be provable but not provable by $\vee$I, it *will* work by this method! So in difficult cases when the goal is a disjunction, it is wise to think about whether you can get one side from the negation of the other. If you can, set up as above. (And reconsider this method, when we get to a simplified version in the extended system *ND+*).

This example was fairly difficult! You may see some longer, but you will not see many harder. The strategies are not a cookbook for performing all derivations — doing derivations remains an art. But the strategies will give you a good start, and take you a long way through the exercises that follow. The theorems immediately below again foreshadow rules of *ND+*.

*T6.19. $\vdash_{ND} \sim(\mathscr{A} \wedge \mathscr{B}) \leftrightarrow (\sim\mathscr{A} \vee \sim\mathscr{B})$

T6.20. $\vdash_{ND} \sim(\mathscr{A} \vee \mathscr{B}) \leftrightarrow (\sim\mathscr{A} \wedge \sim\mathscr{B})$

T6.21. $\vdash_{ND} (\sim\mathscr{A} \rightarrow \mathscr{B}) \leftrightarrow (\mathscr{A} \vee \mathscr{B})$

T6.22. $\vdash_{ND} (\mathscr{A} \rightarrow \mathscr{B}) \leftrightarrow (\sim\mathscr{A} \vee \mathscr{B})$

T6.23. $\vdash_{ND} [\mathscr{A} \wedge (\mathscr{B} \vee \mathscr{C})] \leftrightarrow [(\mathscr{A} \wedge \mathscr{B}) \vee (\mathscr{A} \wedge \mathscr{C})]$

T6.24. $\vdash_{ND} [\mathscr{A} \vee (\mathscr{B} \wedge \mathscr{C})] \leftrightarrow [(\mathscr{A} \vee \mathscr{B}) \wedge (\mathscr{A} \vee \mathscr{C})]$

T6.25. $\vdash_{ND} (\mathscr{A} \leftrightarrow \mathscr{B}) \leftrightarrow [(\mathscr{A} \to \mathscr{B}) \wedge (\mathscr{B} \to \mathscr{A})]$

T6.26. $\vdash_{ND} (\mathscr{A} \leftrightarrow \mathscr{B}) \leftrightarrow [(\mathscr{A} \wedge \mathscr{B}) \vee (\sim\mathscr{A} \wedge \sim\mathscr{B})]$

E6.17. Each of the following begins with a simple application of ~I or ~E for SG4 or SG5. Complete the derivations, and *explain* your use of strategies for a contradiction. Hint: Each of the strategies for a contradiction is used at least once.

*a.

1.	$A \wedge B$	P
2.	$\sim(A \wedge C)$	P
3.	$\quad C$	A (c, ~I)
	$\quad \bot$	
	$\sim C$	

b.

1.	$(\sim B \vee \sim A) \to D$	P
2.	$C \wedge \sim D$	P
3.	$\quad \sim B$	A (c, ~E)
	$\quad \bot$	
	B	

c.

1.	$A \wedge B$	P
2.	$\quad \sim A \vee \sim B$	A (c, ~I)
	$\quad \bot$	
	$\sim(\sim A \vee \sim B)$	

d.

1.	$A \leftrightarrow \sim A$	P
2.	$\quad B$	A (c, ~I)
	$\quad \bot$	
	$\sim B$	

e.

1.	$\sim(A \to B)$	P
2.	$\quad \sim A$	A (c, $\sim$E)
	$\quad \bot$	
	A	

E6.18. Produce derivations to show each of the following. No worked out answers are provided. However, if you get stuck, you will find strategy hints in the back.

*a. $A \to \sim(B \wedge C), B \to C \vdash_{ND} A \to \sim B$

*b. $\vdash_{ND} \sim(A \to A) \to A$

*c. $A \vee B \vdash_{ND} \sim(\sim A \wedge \sim B)$

*d. $\sim(A \wedge B), \sim(A \wedge \sim B) \vdash_{ND} \sim A$

*e. $\vdash_{ND} A \vee \sim A$

*f. $\vdash_{ND} A \vee (A \to B)$

*g. $A \vee \sim B, \sim A \vee \sim B \vdash_{ND} \sim B$

*h. $A \leftrightarrow (\sim B \vee C), B \to C \vdash_{ND} A$

*i. $A \leftrightarrow B \vdash_{ND} (C \leftrightarrow A) \leftrightarrow (C \leftrightarrow B)$

*j. $A \leftrightarrow \sim(B \leftrightarrow \sim C), \sim(A \vee B) \vdash_{ND} C$

*k. $[C \vee (A \vee B)] \wedge (C \to E), A \to D, D \to \sim A \vdash_{ND} C \vee B$

*l. $\sim(A \to B), \sim(B \to C) \vdash_{ND} \sim D$

*m. $C \to \sim A, \sim(B \wedge C) \vdash_{ND} (A \vee B) \to \sim C$

*n. $\sim(A \leftrightarrow B) \vdash_{ND} \sim A \leftrightarrow B$

*o. $A \leftrightarrow B, B \leftrightarrow \sim C \vdash_{ND} \sim(A \leftrightarrow C)$

*p. $A \vee B, \sim B \vee C, \sim C \vdash_{ND} A$

*q. $(\sim A \vee C) \vee D, D \to \sim B \vdash_{ND} (A \wedge B) \to C$

*r. $A \vee D, \sim D \leftrightarrow (E \vee C), (C \wedge B) \vee [C \wedge (F \to C)] \vdash_{ND} A$

*s. $(A \vee B) \vee (C \wedge D), (A \leftrightarrow E) \wedge (B \rightarrow F), G \leftrightarrow \sim(E \vee F), C \rightarrow B \vdash_{ND} \sim G$

*t. $(A \vee B) \wedge \sim C, \sim C \rightarrow (D \wedge \sim A), B \rightarrow (A \vee E) \vdash_{ND} E \vee F$

*E6.19. Produce derivations to demonstrate each of T6.19 - T6.26.

E6.20. Produce derivations to show each of the following. These are particularly challenging. If you can get them, you are doing very well! (In keeping with the spirit of the challenge, no help is provided in the back of the book.)

a. $A \leftrightarrow (B \leftrightarrow C) \vdash_{ND} (A \leftrightarrow B) \leftrightarrow C$

b. $(A \vee B) \rightarrow (A \vee C) \vdash_{ND} A \vee (B \rightarrow C)$

c. $A \rightarrow (B \vee C) \vdash_{ND} (A \rightarrow B) \vee (A \rightarrow C)$

d. $(A \leftrightarrow B) \leftrightarrow (C \leftrightarrow D) \vdash_{ND} (A \leftrightarrow C) \rightarrow (B \rightarrow D)$

e. $\sim(A \leftrightarrow B), \sim(B \leftrightarrow C), \sim(C \leftrightarrow A) \vdash_{ND} \sim K$

E6.21. For each of the following, produce a good translation including interpretation function. Then use a derivation to show that the argument is valid in *ND*. The first two are suggested from the history of philosophy; the last is our familiar case from p. 2.

a. We have knowledge about numbers.
If Platonism is true, then numbers are not in spacetime.
Either numbers are in spacetime, or we do not interact with them.
We have knowledge about numbers only if we interact with them.

Platonism is not true.

b. There is evil
If god is good, there is no evil unless he has an excuse for allowing it.
If god is omnipotent, then he does not have an excuse for allowing evil.

God is not both good and omnipotent.

c. If Bob goes to the fair, then so do Daniel and Edward. Albert goes to the fair only if Bob or Carol go. If Daniel goes, then Edward goes only if Fred goes. But not both Fred and Albert go. So Albert goes to the fair only if Carol goes too.

d. If I think dogs fly, then I am insane or they have really big ears. But if dogs do not have really big ears, then I am not insane. So either I do not think dogs fly, or they have really big ears.

e. If the maid did it, then it was done with a revolver only if it was done in the parlor. But if the butler is innocent, then the maid did it unless it was done in the parlor. The maid did it only if it was done with a revolver, while the butler is guilty if it did happen in the parlor. So the butler is guilty.

E6.22. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. Derivations as games, and the condition on rules.

b. Accessibility, and auxiliary assumptions.

c. The rules $\vee$I and $\vee$E.

d. The strategies for a goal.

e. The strategies for a contradiction.

6.3 Quantificational

Our full system *ND* includes all the rules for the sentential part of *ND*, along with I- and E-rules for $\forall$ and $\exists$ and for equality. After some quick introductory remarks, we will take up the new rules, and say a bit about strategy.

First, we do not sacrifice any of the rules we have so far. Recall that our rules apply to formulas of quantificational languages as well as to formulas of sentential ones. Thus, for example, $Fx \rightarrow \forall xFx$ and Fx are of the form $\mathcal{P} \rightarrow \mathcal{Q}$ and $\mathcal{P}$. So we might move from them to $\forall xFx$ by $\rightarrow$E as before. And similarly for other rules. Here is a short example.

(AR)

1.	$\forall xFx \wedge \exists x\forall y(Hx \vee Zy)$	P
2.	$\quad Kx$	A (g, $\rightarrow$I)
3.	$\quad \forall xFx$	1 $\wedge$E
4.	$Kx \rightarrow \forall xFx$	2-3 $\rightarrow$I

The goal is of the form $\mathcal{P} \to \mathcal{Q}$; so we set up to get it in the usual way. And the subderivation is particularly simple. Notice that formulas of the sort $\forall x(Kx \to Fx)$ and Kx are *not* of the form $\mathcal{P} \to \mathcal{Q}$ and $\mathcal{P}$. The main operator of $\forall x(Kx \to Fx)$ is $\forall x$, not $\to$. So $\to$E does not apply. That is why we need new rules for the quantificational operators.

For our quantificational rules, we need a couple of notions already introduced in chapter 3. Again, for any formula $\mathcal{A}$, variable x, and term t, say $\mathcal{A}^x_t$ is $\mathcal{A}$ with all the free instances of x replaced by t. And t is *free for x in $\mathcal{A}$* iff all the variables in the replacing instances of t remain free after substitution in $\mathcal{A}^x_t$. Thus, for example,

(AS) $(\forall xRxy \vee Px)^x_y$ is $\forall xRxy \vee Py$

There are three instances of x in $\forall xRxy \vee Px$, but only the last is free; so y is substituted only for that instance. Since the substituted y is free in the resultant expression, y is free for x in $\forall xRxy \vee Px$. Similarly,

(AT) $[\forall x(x = y) \vee Ryx]^y_{f^1x}$ is $\forall x(x = f^1x) \vee Rf^1xx$

Both instances of y in $\forall x(x = y) \vee Ryx$ are free; so our substitution replaces both. But the x in the first instance of f^1x is bound upon substitution; so f^1x is not free for y in $\forall x(x = y) \vee Ryx$. Notice that if x is not free in $\mathcal{A}$, then replacing every free instance of x in $\mathcal{A}$ with some term results in no change. So if x is not free in $\mathcal{A}$, then $\mathcal{A}^x_t$ is $\mathcal{A}$. Similarly, $\mathcal{A}^x_x$ is just $\mathcal{A}$ itself. Further, any variable x is sure to be free for itself in a formula $\mathcal{A}$ — if every *free* instance of variable x is "replaced" with x, then the replacing instances are sure to be free! And constants are sure to be free for a variable x in a formula $\mathcal{A}$. Since a constant c is a term without variables, no variable in the replacing term is bound upon substitution for free instances of x.

With this said, we are ready to turn to our rules. We begin with the easier ones, and work from there.

6.3.1 ∀E and ∃I

∀E and ∃I are straightforward. For the former, for any variable x, given an accessible formula $\forall x\mathcal{P}$ on line a, if term t is free for x in $\mathcal{P}$, one may move to $\mathcal{P}^x_t$ with justification, a ∀E.

∀E	a.	$\forall x\mathcal{P}$		provided t is free for x in $\mathcal{P}$
		$\mathcal{P}^x_t$	a ∀E	

∀E removes a quantifier, and substitutes a term t for resulting free instances of x, so long as t is free in the resulting formula. Observe that t is *always* free if it is a constant, or a variable that does not appear at all in $\mathcal{P}$. We sometimes say that variable x is *instantiated* by term t. Thus, for example, $\forall x \exists y Lxy$ is of the form $\forall x \mathcal{P}$, where $\mathcal{P}$ is $\exists y Lxy$. So by ∀E we can move from $\forall x \exists y Lxy$ to $\exists y Lay$, removing the quantifier and substituting a for x. And similarly, since the complex terms $f^1 a$ and $g^2 zb$ are free for x in $\exists y Lxy$, ∀E legitimates moving from $\forall x \forall y Lxy$ to $\exists y L f^1 ay$ or $\exists y L g^2 zby$. What we cannot do is move from $\forall x \exists y Lxy$ to $\exists y Lyy$ or $\exists y L f^1 yy$. These violate the constraint insofar as a variable of the substituted term is bound by a quantifier in the resulting formula.

Intuitively, the motivation for this rule is clear: If $\mathcal{P}$ is satisfied for *every* assignment to variable x, then it is sure to be satisfied for the thing assigned to t, whatever that thing may be. Thus, for example, if everyone loves someone, $\forall x \exists y Lxy$, it is sure to be the case that Al, and Al's father love someone — that $\exists y Lay$ and $\exists y L f^1 ay$. But from everyone loves someone, it does not follow that anyone loves themselves, that $\exists y Lyy$, or that anyone is loved by their father $\exists y L f^1 yy$. Though we know Al and Al's father loves someone, we do not know who that someone might be. We therefore require that the replacing term be independent of quantifiers in the rest of the formula.

Here are some examples. Notice that we continue to apply bottom-up goal-oriented thinking.

(AU)

1.	$\forall x \forall y Hxy$	P
2.	$Hcf^2 ab \rightarrow \forall z Kz$	P
3.	$\forall y Hcy$	1 ∀E
4.	$Hcf^2 ab$	3 ∀E
5.	$\forall z Kz$	2,4 →E
6.	Kb	5 ∀E

Our original goal is Kb. We could get this by ∀E if we had $\forall z Kz$. So we set that as a subgoal. This leads to $Hcf^2 ab$ as another subgoal. And we get this from (1) by two applications of ∀E. The constant c is free for x in $\forall y Hxy$ so we move from $\forall x \forall y Hxy$ to $\forall y Hcy$ by ∀E. And the complex term $f^2 ab$ is free for y in Hcy, so we move from $\forall y Hcy$ to $Hcf^2 ab$ by ∀E. And similarly, we get Kb from $\forall z Kz$ by ∀E.

Here is another example, also illustrating strategic thinking.

(AV)

1.	$\forall x Bx$	P
2.	$\forall x(Cx \rightarrow \sim Bx)$	P
3.	$\quad Ca$	A (c, $\sim$I)
4.	$\quad Ca \rightarrow \sim Ba$	2 $\forall$E
5.	$\quad \sim Ba$	4,3 $\rightarrow$E
6.	$\quad Ba$	1 $\forall$E
7.	$\quad \perp$	6,5 $\perp$I
8.	$\sim Ca$	3-7 $\sim$I

Our original goal is $\sim Ca$; so we set up to get it by $\sim$I. And our contradiction appears at the level of atomics and negated atomics. The constant a is free for x in $Cx \rightarrow \sim Bx$. So we move from $\forall x(Cx \rightarrow \sim Bx)$ to $Ca \rightarrow \sim Ba$ by $\forall$E. And similarly, we move from $\forall x Bx$ to Ba by $\forall$E. Notice that we could use $\forall$E to instantiate the universal quantifiers to *any* terms. We pick the constant a because it does us some good in the context of our assumption Ca — itself driven by the goal, $\sim Ca$. And it is typical to "swoop" in with universal quantifiers to put variables on terms that matter in a given context.

$\exists$I is equally straightforward. For variable x, given an accessible formula $\mathcal{P}^x_t$ on line a, where term t is free for x in formula $\mathcal{P}$, one may move to $\exists x\mathcal{P}$, with justification, a $\exists$I.

$\exists$I

a.	$\mathcal{P}^x_t$		provided t is free for x in $\mathcal{P}$
	$\exists x\mathcal{P}$	a $\exists$I	

The statement of this rule is somewhat in reverse from the way one expects it to be: Supposing that t is free for x in $\mathcal{P}$, when one removes the quantifier from the *result* and replaces every free instance of x with t one ends up with the *start*. A consequence is that one starting formula might legitimately lead to different results by $\exists$I. Thus if $\mathcal{P}$ is any of Fxx, Fxa, or Fax, then $\mathcal{P}^x_a$ is Faa. So $\exists$I allows a move from Faa to any of $\exists xFxx$, $\exists xFax$ or $\exists xFxa$. In doing a derivation, there is a sense in which we replace one or more instances of a in Faa with x, and add the quantifier to get the result. But then notice that not every instance of the term need be replaced. Officially the rule is stated the other way: Removing the quantifier from the result, and replacing free instances of the variable, yields the initial formula. Be clear about this in your mind. The requirement that t be free for x in $\mathcal{P}$ prevents moving from $\forall yLyy$ or $\forall yLf^1yy$ to $\exists x\forall yLxy$. The term from which we generalize must be free in the sense that it has no bound variable!

Again, the motivation for this rule is clear. If $\mathcal{P}$ is satisfied for the individual assigned to t, it is sure to be satisfied for *some* individual. Thus, for example, if Al or

Al's father love everyone, $\forall yLay$ or $\forall yLf^1ay$, it is sure to be the case that someone loves everyone $\exists x\forall yLxy$. But from the premise that everyone loves themselves $\forall yLyy$, or that everyone is loved by their father $\forall yLf^1yy$ it does not follow that someone loves everyone. Again, the constraint on the rule requires that the term on which we generalize be independent of quantifiers in the rest of the formula.

Here are a couple of examples. The first is relatively simple. The second illustrates the "duality" between $\forall$E and $\exists$I.

(AW)

1.	Ha	P
2.	$\exists yHy \to \forall xJx$	P
3.	$\exists yHy$	1 $\exists$I
4.	$\forall xJx$	2,3 $\to$E
5.	Ja	4 $\forall$E
6.	$Ha \wedge Ja$	1,5 $\wedge$I
7.	$\exists x(Hx \wedge Jx)$	6 $\exists$I

$Ha \wedge Ja$ is $(Hx \wedge Jx)^x_a$ so we can get $\exists x(Hx \wedge Jx)$ from $Ha \wedge Ja$ by $\exists$I. Ha is already a premise, so we set Ja as a subgoal. Ja comes by $\forall$E from $\forall xJx$, and to get this we set $\exists yHy$ as another subgoal. And $\exists yHy$ follows directly by $\exists$I from Ha. Observe that, for now, the natural way to produce a formula with main operator $\exists$ is by $\exists$I. You should fold this into your strategic thinking.

For the second example recall, from translations, that $\sim\forall x\sim\mathcal{P}$ is equivalent to $\exists x\mathcal{P}$, and $\sim\exists x\sim\mathcal{P}$ is equivalent to $\forall x\mathcal{P}$. Given this, it turns out that we can use the universal rule with an effect something like $\exists$I, and the existential rule with an effect like $\forall$E. The following pair of derivations illustrate this point.

(AX)

1.	Pa	P
2.	$\quad \forall x\sim Px$	A (c, $\sim$I)
3.	$\quad \sim Pa$	2 $\forall$E
4.	$\quad \bot$	1,3 $\bot$I
5.	$\sim\forall x\sim Px$	2-4 $\sim$I

(AY)

1.	$\sim\exists x\sim Px$	P
2.	$\quad \sim Pa$	A (c, $\sim$E)
3.	$\quad \exists x\sim Px$	2 $\exists$I
4.	$\quad \bot$	3,1 $\bot$I
5.	Pa	2-4 $\sim$E

By $\exists$I we could move from Pa to $\exists xPx$ in one step. In (AX) we use the universal rule to move from the same premise to the equivalent $\sim\forall x\sim Px$. Indeed, $\exists xPx$ abbreviates this very expression. Similarly, by $\forall$E we could move from $\forall xPx$ to Pa in one step. In (AY), we move to the same result by the existential rule from the equivalent $\sim\exists x\sim Px$. Thus there is a sense in which, in the presence of rules for negation, the work done by one of these quantifier rules is very similar to, or can substitute for, the work done by the other.

E6.23. Complete the following derivations by filling in justifications for each line. Then for each application of $\forall$E or $\exists$I, show that the "free for" constraint is met. Hint: it may be convenient to xerox the problems, and fill in your answers directly on the copy.

a.
1. $\forall x(Ax \to Bxf^1x)$
2. $\forall xAx$
3. Af^1c
4. $Af^1c \to Bf^1cf^1f^1c$
5. $Bf^1cf^1f^1c$

*b.
1. Gaa
2. $\exists yGay$
3. $\exists x\exists yGxy$

c.
1. $\forall x(Rx \wedge Jx)$
2. $Rk \wedge Jk$
3. Rk
4. Jk
5. $Jk \wedge Rk$
6. $\exists y(Jy \wedge Ry)$

d.
1. $\exists x(Rx \wedge Gx) \to \forall yFy$
2. $\forall zGz$
3. Ra
4. Ga
5. $Ra \wedge Ga$
6. $\exists x(Rx \wedge Gx)$
7. $\forall yFy$
8. Fg^2ax

e.
1. $\sim\exists zFg^1z$
2. $\quad\forall xFx$
3. $\quad Fg^1k$
4. $\quad\exists zFg^1z$
5. $\quad\bot$
6. $\sim\forall xFx$

E6.24. The following are not legitimate *ND* derivations. In each case, explain why.

a.

1.	$\forall xFx \leftrightarrow Gx$	P
2.	$Fj \leftrightarrow Gj$	1 $\forall$E

*b.

1.	$\forall x\exists yGxy$	P
2.	$\exists yGyy$	1 $\forall$E

c.

1.	$\forall y(Fay \rightarrow Gy)$	P
2.	$Fay \rightarrow Gf^1b$	1 $\forall$E

d.

1.	$\forall yGf^2xyy$	P
2.	$\exists x\forall yGxy$	1 $\exists$I

e.

1.	Gj	P
2.	$\exists xGf^1x$	1 $\exists$I

E6.25. Provide derivations to show each of the following.

a. $\forall xFx \vdash_{ND} Fa \wedge Fb$

*b. $\forall x\forall yFxy \vdash_{ND} Fab \wedge Fba$

c. $\forall x(Gf^1x \rightarrow \forall yAyx), Gf^1b \vdash_{ND} Af^1cb$

d. $\forall x\forall y(Hxy \rightarrow Dyx), {\sim}Dab \vdash_{ND} {\sim}Hba$

e. $\vdash_{ND} [\forall x\forall yFxy \wedge \forall x(Fxx \rightarrow A)] \rightarrow A$

f. $Fa, Ga \vdash_{ND} \exists x(Fx \wedge Gx)$

*g. $Gaf^1z \vdash_{ND} \exists x\exists yGxy$

h. $\vdash_{ND} (Fa \vee Fb) \rightarrow \exists xFx$

i. $Gaa \vdash_{ND} \exists x\exists y(Kxx \rightarrow Gxy)$

j. $\forall xFx, Ga \vdash_{ND} \exists y(Fy \wedge Gy)$

*k. $\forall x(Fx \rightarrow Gx), \exists yGy \rightarrow Ka \vdash_{ND} Fa \rightarrow \exists xKx$

l. $\forall x\forall yHxy \vdash_{ND} \exists y\exists xHyx$

m. $\forall x({\sim}Bx \rightarrow Kx), {\sim}Kf^1x \vdash_{ND} Bf^1x$

n. $\forall x\forall y(Fxy \rightarrow {\sim}Fyx) \vdash_{ND} \exists z{\sim}Fzz$

o. $\forall x(Fx \rightarrow Gx), Fa \vdash_{ND} \exists x({\sim}Gx \rightarrow Hx)$

6.3.2 ∀I and ∃E

In parallel with ∀E and ∃I, rules for ∀I and ∃E are a linked pair. ∀I is as follows: For variables v and x, given an accessible formula $\mathcal{P}^x_v$ at line a, where v is free for x in $\mathcal{P}$, v is not free in any undischarged assumption, and v is not free in $\forall x\mathcal{P}$, one may move to $\forall x\mathcal{P}$ with justification a ∀I.

∀I	a. $\mathcal{P}^x_v$		provided (i) v is free for x in $\mathcal{P}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\forall x\mathcal{P}$
	$\forall x\mathcal{P}$	a ∀I	

The form of this rule is like a constrained ∃I when t is a variable: from $\mathcal{P}^x_v$ we move to the quantified expression $\forall x\mathcal{P}$. The underlying difference is in the special constraints. First, the combination of (i) and (iii) require that v and x appear free in just the same places. If v is free for x in $\mathcal{P}$, then v is free in $\mathcal{P}^x_v$ everywhere x is free in $\mathcal{P}$; if v is not free in $\forall x\mathcal{P}$, then v is free in $\mathcal{P}^x_v$ only where x is free in $\mathcal{P}$. So you get back-and-forth between $\mathcal{P}$ and $\mathcal{P}^x_v$ by replacing every free x with v or every free v with x. This two-way requirement is not present for ∃I.

In addition, v cannot be free in an auxiliary assumption still in effect when ∀I is applied. Recall that a formula is true when it is satisfied on any variable assignment. As it turns out (and we shall see in detail in Part II), the truth of a formula with a free variable therefore implies the truth of its universal quantification. But this is not so under the scope of an assumption in which the variable is free. Under the scope of an assumption with a free variable, we effectively *constrain* the range of assignments under consideration to ones where the assumption is satisfied. Thus under any such assumption, the move to a universal quantification is not justified. For the universal quantification to be justified, the formula must be satisfied for *any* assignment to v, and when v is free in an undischarged assumption we do not have that guarantee. Only when assignments to v are arbitrary, when reasoning with respect to v might apply to any individual, is the move from $\mathcal{P}^x_v$ to $\forall x\mathcal{P}$ justified. Again, observe that no such constraint is required for ∃I, which depends on satisfaction for just a single individual, so that any assignment and term will do.

Once you get your mind around them, these constraints are not difficult. Somehow, though, managing them is a common source of frustration for beginning students. However, there is a simple way to be sure that the constraints are met. Suppose you have been following the strategies, along the lines from before, and come to a goal of the sort, $\forall x\mathcal{P}$. It is natural to expect to get this by ∀I from $\mathcal{P}^x_v$. You will be sure to satisfy the constraints, if you set $\mathcal{P}^x_v$ as a subgoal, *where v does not appear elsewhere in the derivation*. If v does not otherwise appear in the derivation, (i) there cannot be any v quantifier in $\mathcal{P}$, so v is sure to be free for x in $\mathcal{P}$. If v does

not otherwise appear in the derivation, (ii) v cannot appear in any assumption, and so be free in an undischarged assumption. And if v does not otherwise appear in the derivation, (iii) it cannot appear at all in $\forall x\mathcal{P}$, and so cannot be free in $\forall x\mathcal{P}$. It is not always *necessary* to use a new variable in order to satisfy the constraints, and sometimes it is possible to simplify derivations by clever variable selection. However, we shall make it our standard procedure to do so.

Here are some examples. The first is very simple, but illustrates the basic idea underlying the rule.

(AZ)

1.	$\forall x(Hx \wedge Mx)$	P
	Hj	
	$\forall yHy$	___ $\forall$I

1.	$\forall x(Hx \wedge Mx)$	P
2.	$Hj \wedge Mj$	1 $\forall$E
3.	Hj	2 $\wedge$E
4.	$\forall yHy$	3 $\forall$I

The goal is $\forall yHy$. So, picking a variable new to the derivation, we set up to get this by $\forall$I from Hj. This goal is easy to obtain from the premise by $\forall$E and $\wedge$E. If every x is such that both Hx and Mx, it is not surprising that every y is such that Hy. The general content from the quantifier is converted to the form with free variables, manipulated by ordinary rules, and converted back to quantified form. This is typical.

Another example has free variables in an auxiliary assumption.

(BA)

1.	$\forall x(Ex \to Sx)$	P
2.	$\forall z(Sz \to Kz)$	P
	$Ej \to Kj$	
	$\forall x(Ex \to Kx)$	___ $\forall$I

1.	$\forall x(Ex \to Sx)$	P
2.	$\forall z(Sz \to Kz)$	P
3.	$\quad Ej$	A (g, $\to$I)
4.	$\quad Ej \to Sj$	1 $\forall$E
5.	$\quad Sj$	4,3 $\to$E
6.	$\quad Sj \to Kj$	2 $\forall$E
7.	$\quad Kj$	6,5 $\to$E
8.	$Ej \to Kj$	3-7 $\to$I
9.	$\forall x(Ex \to Kx)$	8 $\forall$I

Given the goal $\forall x(Ex \to Kx)$, we immediately set up to get it by $\forall$I from $Ej \to Kj$. At that stage, j does not appear elsewhere in the derivation, and we can therefore be sure that the constraints will be met when it comes time to apply $\forall$I. The derivation is completed by the usual strategies. Observe that j appears in an auxiliary assumption at (3). This is no problem insofar as the assumption is discharged by the time $\forall$I is applied. We would not, however, be able to conclude, say, $\forall xSx$ or $\forall xKx$ inside the subderivation, since at that stage, the variable j is free in the undischarged assumption. But, of course, given the strategies, there should be no temptation whatsoever to do so! For when we set up for $\forall$I, we set up to do it in a way that is sure to satisfy the constraints.

A last example introduces multiple quantifiers and, again, emphasizes the importance of following the strategies. Insofar as the conclusion merely exchanges variables with the premise, it is no surprise that there is a way for it to be done.

(BB)

1.	$\forall x(Gx \rightarrow \forall yFyx)$	P
	$Gj \rightarrow \forall xFxj$	
	$\forall y(Gy \rightarrow \forall xFxy)$	_ $\forall$I

1.	$\forall x(Gx \rightarrow \forall yFyx)$	P
2.	$\quad Gj$	A (g, $\rightarrow$I)
	$\quad Fkj$	
	$\quad \forall xFxj$	_ $\forall$I
	$Gj \rightarrow \forall xFxj$	2-_ $\rightarrow$I
	$\forall y(Gy \rightarrow \forall xFxy)$	_ $\forall$I

First, we set up to get $\forall y(Gy \rightarrow \forall xFxy)$ from $Gj \rightarrow \forall xFxj$. The variable j does not appear in the derivation, so we expect that the constraints on $\forall$I will be satisfied. But our new goal is a conditional, so we set up to go for it by $\rightarrow$I in the usual way. This leads to $\forall xFxj$ as a goal, and we set up to get it from Fkj, where k does not otherwise appear in the derivation. Observe that we have at this stage an undischarged assumption in which j appears. However, our plan is to generalize on k. Since k is new at this stage, we are fine. Of course, this assumes that we are following the strategies so that our new variable automatically avoids variables free in assumptions under which this instance of $\forall$I falls. This goal is easily obtained and the derivation completed as follows.

1.	$\forall x(Gx \rightarrow \forall yFyx)$	P
2.	$\quad Gj$	A (g, $\rightarrow$I)
3.	$\quad Gj \rightarrow \forall yFyj$	1 $\forall$E
4.	$\quad \forall yFyj$	3,2 $\rightarrow$E
5.	$\quad Fkj$	4 $\forall$E
6.	$\quad \forall xFxj$	5 $\forall$I
7.	$Gj \rightarrow \forall xFxj$	2-6 $\rightarrow$I
8.	$\forall y(Gy \rightarrow \forall xFxy)$	7 $\forall$I

When we apply $\forall$I the first time, we replace each instance of k with x and add the x quantifier. When we apply $\forall$I the second time, we replace each instance of j with y and add the y quantifier. This is just how we planned for the rules to work.

$\exists$E appeals to both a formula and a subderivation. For variables $\mathscr{v}$ and $\mathscr{x}$, given an accessible formula $\exists \mathscr{x}\mathscr{P}$ at a, and an accessible subderivation beginning with $\mathscr{P}^{\mathscr{x}}_{\mathscr{v}}$ at b and ending with $\mathscr{Q}$ against its scope line at c — where $\mathscr{v}$ is free for $\mathscr{x}$ in $\mathscr{P}$, $\mathscr{v}$ is free in no undischarged assumption, $\mathscr{v}$ is not free in $\exists \mathscr{x}\mathscr{P}$ or in $\mathscr{Q}$, one may move to $\mathscr{Q}$, with justification *a,b-c* $\exists$E.

∃E

$$\begin{array}{r|l|l}
\text{a.} & \exists x\mathcal{P} & \\
\text{b.} & \quad \mid \mathcal{P}^{x}_{v} & \text{A } (g, a\exists\text{E}) \\
 & \quad \mid & \\
\text{c.} & \quad \mid \mathcal{Q} & \\
 & \mathcal{Q} & \text{a,b-c } \exists\text{E}
\end{array}$$

provided (i) v is free for x in $\mathcal{P}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\exists x\mathcal{P}$ or in $\mathcal{Q}$

Notice that the assumption comes with an exit strategy as usual. We can think of this rule on analogy with ∨E. A universally quantified expression is something like a big conjunction: if $\forall x\mathcal{P}$, then this element of U is $\mathcal{P}$ *and* that element of U is $\mathcal{P}$ *and* .... And an existentially quantified expression is something like a big disjunction: if $\exists x\mathcal{P}$, then this element of U is $\mathcal{P}$ *or* that element of U is $\mathcal{P}$ *or* .... What we need to show is that no matter which thing happens to be the one that is $\mathcal{P}$, we get the result that $\mathcal{Q}$. Given this, we are in a position to conclude that $\mathcal{Q}$. As for the case of ∀I, then, the constraints guarantee that our reasoning applies to any individual.

Again, if you are following the strategies, a simple way to guarantee that the constraints are met is to use a variable new to the derivation for the assumption. Suppose you are going for goal $\mathcal{Q}$. In parallel with ∨, when presented with an accessible formula with main operator ∃, it is wise to go for the entire goal by ∃E.

(BC)

$$\begin{array}{r|l|l}
\text{a.} & \exists x\mathcal{P} & \\
 & & \\
 & & \\
 & \mathcal{Q} & \text{(goal)}
\end{array}
\qquad\qquad
\begin{array}{r|l|l}
\text{a.} & \exists x\mathcal{P} & \\
\text{b.} & \quad \mid \mathcal{P}^{x}_{v} & \text{A } (g, a\exists\text{E}) \\
 & \quad \mid & \\
\text{c.} & \quad \mid \mathcal{Q} & \text{(goal)} \\
 & \mathcal{Q} & \text{a,b-c } \exists\text{E}
\end{array}$$

If v does not otherwise appear in the derivation, then (i) there is no v quantifier in $\mathcal{P}$ and v is sure to be free for x in $\mathcal{P}$. If v does not otherwise appear in the derivation (ii) v does not appear in any other assumption and so is not free in any undischarged auxiliary assumption. And if v does not otherwise appear in the derivation (iii) v does not appear in either $\exists x\mathcal{P}$ or in $\mathcal{Q}$ and so is not free in $\exists x\mathcal{P}$ or in $\mathcal{Q}$. Thus we adopt the same simple expedient to guarantee that the constraints are met. Of course, this presupposes we are following the strategies enough so that other assumptions are in place when we make the assumption for ∃E, and that we are clear about the exit strategy, so that we know what $\mathcal{Q}$ will be! The variable is new relative to this much setup.

Here are some examples. The first is particularly simple, and should seem intuitively right. Notice again, that given an accessible formula with main operator ∃, we go directly for the goal by ∃E.

(BD)

1.	$\exists x(Fx \wedge Gx)$	P
2.	$\quad Fj \wedge Gj$	A (g, 1$\exists$E)
	$\quad \exists xFx$	
	$\exists xFx$	1, __ $\exists$E

1.	$\exists x(Fx \wedge Gx)$	P
2.	$\quad Fj \wedge Gj$	A (g, 1$\exists$E)
3.	$\quad Fj$	2 $\wedge$E
4.	$\quad \exists xFx$	3 $\exists$I
5.	$\exists xFx$	1,2-4 $\exists$E

Given an accessible formula with main operator $\exists$, we go for the goal by $\exists$E. This gives us a subderivation with the same goal, and our assumption with the new variable. As it turns out, this goal is easy to obtain, with instances of $\wedge$E and $\exists$I. We could not do $\forall$I to introduce $\forall xFx$ under the scope of the assumption with j free. But $\exists$I is *not* so constrained. So we complete the derivation as above. If some x is such that both Fx and Gx then of course some x is such that Fx. Again, we are able to take the quantifier off, manipulate the expressions with free variables, and put the quantifier back on.

Observe that the following is a mistake. It violates the third constraint that v the variable to which we instantiate the existential, is not free in $\mathcal{Q}$ the formula that results from $\exists$E.

(BE)

1.	$\exists x(Fx \wedge Gx)$	P	
2.	$\quad Fj \wedge Gj$	A (g, 1$\exists$E)	
3.	$\quad Fj$	2 $\wedge$E	
4.	Fj	1,2-3 $\exists$E	Mistake!
5.	$\exists xFx$	4 $\exists$I	

If you are following the strategies, there should be no temptation to do this. In the above example (BD), we go for the *goal* $\exists xFx$ by $\exists$E. At that stage, the variable of the assumption j is new to the derivation and so does not appear in the goal. So all is well. This case (BE) does not introduce a variable that is new relative to the goal of the subderivation, and so runs into trouble.

Very often, a goal from $\exists$E is existentially quantified — for introducing an existential quantifier is one way of eliminating the variable from the assumption so that it is not free in the goal. In fact, we do not have to think much about this, insofar as we explicitly introduce the assumption by a variable not in the goal. However, it is not always the case that the goal for $\exists$E is existentially quantified. Here is a simple case of that sort.

(BF)

1.	$\exists xFx$	P	1.	$\exists xFx$	P
2.	$\forall z(\exists yFy \to Gz)$	P	2.	$\forall z(\exists yFy \to Gz)$	P
3.	Fj	A (g, 1$\exists$E)	3.	Fj	A (g, 1$\exists$E)
			4.	$\exists yFy \to Gk$	2 $\forall$E
			5.	$\exists yFy$	3 $\exists$I
			6.	Gk	4,5 $\to$E
	$\forall xGx$		7.	$\forall xGx$	6 $\forall$I
	$\forall xGx$	1, __ $\exists$E	8.	$\forall xGx$	1,3-7 $\exists$E

Again, given an existential premise, we set up to reach the goal by ∃E, where the variable in the assumption is new. In this case, the goal is universally quantified, and illustrates the point that any formula may be the goal for ∃E. In this case, we reach the goal in the usual way. To reach $\forall xGx$ set Gk as goal; at this stage, k is new to the derivation, and so not free in any undischarged assumption. So there is no problem about ∀I. Then it is a simple matter of exploiting accessible lines for the result.

Here is an example with multiple quantifiers. It is another case which makes sense insofar as the premise and conclusion merely exchange variables.

(BG)

1.	$\exists x(Fx \wedge \exists yGxy)$	P	1.	$\exists x(Fx \wedge \exists yGxy)$	P
2.	$Fj \wedge \exists yGjy$	A (g 1$\exists$E)	2.	$Fj \wedge \exists yGjy$	A (g, 1$\exists$E)
			3.	$\exists yGjy$	2 $\wedge$E
			4.	Gjk	A (g, 3$\exists$E)
				$\exists y(Fy \wedge \exists xGyx)$	
	$\exists y(Fy \wedge \exists xGyx)$			$\exists y(Fy \wedge \exists xGyx)$	3, 4- __ $\exists$E
	$\exists y(Fy \wedge \exists xGyx)$	1, 2- __ $\exists$E		$\exists y(Fy \wedge \exists xGyx)$	1, 2- __ $\exists$E

The premise is an existential, so we go for the goal by ∃E. This gives us the first subderivation, with the same goal, and new variable j substituted for x. But just a bit of simplification gives us another existential on line (3). Thus, following the standard strategies, we set up to go for the goal again by ∃E. At this stage, j is no longer new, so we set up another subderivation with new variable k substituted for y. Now the derivation is reasonably straightforward.

1.	$\mid \exists x(Fx \wedge \exists yGxy)$	P
2.	$\mid\;\mid Fj \wedge \exists yGjy$	A (g, 1$\exists$E)
3.	$\mid\;\mid \exists yGjy$	2 $\wedge$E
4.	$\mid\;\mid\;\mid Gjk$	A (g, 3$\exists$E)
5.	$\mid\;\mid\;\mid \exists xGjx$	4 $\exists$I
6.	$\mid\;\mid\;\mid Fj$	2 $\wedge$E
7.	$\mid\;\mid\;\mid Fj \wedge \exists xGjx$	6,5 $\wedge$I
8.	$\mid\;\mid\;\mid \exists y(Fy \wedge \exists xGyx)$	7 $\exists$I
9.	$\mid\;\mid \exists y(Fy \wedge \exists xGyx)$	3, 4-8 $\exists$E
10.	$\mid \exists y(Fy \wedge \exists xGyx)$	1, 2-9 $\exists$E

$\exists$I applies in the scope of the subderivations. And we put Fj and $\exists xGjx$ together so that the outer quantifier goes on properly, with y in the right slots.

Finally, observe that $\forall$I and $\exists$I also constitute a dual to one another. The derivations to show this are relatively difficult. But to not worry about that. It is enough to understand the steps. For the parallel to $\forall$I, suppose the constraints are met for a derivation of $\forall xPx$ from Pj. And for the parallel to $\exists$E, suppose it is possible to derive Q by $\exists$E from $\exists xPx$; so from application of that rule, in a subderivation, we can get Q from Pj.

(BH)

1.	$\mid Pj$	P
2.	$\mid\;\mid \exists x{\sim}Px$	A (c, $\sim$I)
3.	$\mid\;\mid\;\mid {\sim}Pj$	A (c, 2$\exists$E)
4.	$\mid\;\mid\;\mid \perp$	1,3 $\perp$I
5.	$\mid\;\mid \perp$	2,3-4 $\exists$E
6.	$\mid {\sim}\exists x{\sim}Px$	2-5 $\sim$I

(BI)

1.	$\mid {\sim}\forall x{\sim}Px$	P
2.	$\mid\;\mid {\sim}Q$	A (c, $\sim$E)
3.	$\mid\;\mid\;\mid Pj$	A (c, $\sim$I)
	$\mid\;\mid\;\mid \vdots$	
4.	$\mid\;\mid\;\mid Q$	(somehow)
5.	$\mid\;\mid\;\mid \perp$	4,2 $\perp$I
6.	$\mid\;\mid {\sim}Pj$	3-5 $\sim$I
7.	$\mid\;\mid \forall x{\sim}Px$	6 $\forall$I
8.	$\mid\;\mid \perp$	7,1 $\perp$I
9.	$\mid Q$	2-8 $\sim$E

Where Pj is a premise, it would be possible to derive $\forall xPx$ in one step by $\forall$I. But in (BH) from the same start, we derive the equivalent ${\sim}\exists x{\sim}Px$ by the existential rule. Since conditions for the universal rule apply, j is not free in an undischarged assumption, is free for x in ${\sim}Px$ and is not free in $\exists x{\sim}Px$. In this case, it matters that $\perp$ abbreviates $Z \wedge {\sim}Z$ and so includes no instance of j. So the constraints are satisfied. Similarly, if it is possible to derive Q by $\exists$E from $\exists xPx$, we would set up a subderivation starting with Pj, derive Q and use $\exists$E to exit with the Q. In (BI) we begin with the equivalent ${\sim}\forall x{\sim}Px$ and, supposing it is possible in a subderivation

to derive Q from Pj, use the universal rule to derive Q. Since conditions for the existential rule apply, j is free for x in $\sim Px$ and not free in $\forall x \sim Px$. Observe also that the assumption Pj is discharged by the time $\forall$I is applied, and that the constraint on $\exists$E requires that j is not free in Q or other undischarged assumptions. Thus, again, there is a sense in which in the presence of rules for negation, the work done by one of these quantifier rules is very similar to, or can substitute for, the work done by the other.

E6.26. Complete the following derivations by filling in justifications for each line. Then for each application of $\forall$I or $\exists$E show that the constraints are met by running through each of the three requirements. Hint: it may be convenient to xerox the problems, and fill in your answers directly on the copy.

a.
1. $\forall x(Hx \to Rx)$
2. $\forall y Hy$
3. $Hj \to Rj$
4. Hj
5. Rj
6. $\forall z Rz$

*b.
1. $\forall y(Fy \to Gy)$
2. $\exists z Fz$
3. $\quad Fj$
4. $\quad Fj \to Gj$
5. $\quad Gj$
6. $\quad \exists x Gx$
7. $\exists x Gx$

c.
1. $\exists x \forall y \forall z Hxyz$
2. $\quad \forall y \forall z Hjyz$
3. $\quad \forall z Hjf^1kz$
4. $\quad Hjf^1kf^1k$
5. $\quad \exists x Hxf^1kf^1k$
6. $\quad \forall y \exists x Hxf^1yf^1y$
7. $\forall y \exists x Hxf^1yf^1y$

d.

1. | $\forall y \forall x(Fx \to By)$
2. | | $\exists x Fx$
3. | | | Fj
4. | | | $\forall x(Fx \to Bk)$
5. | | | $Fj \to Bk$
6. | | | Bk
7. | | Bk
8. | $\exists x Fx \to Bk$
9. | $\forall y(\exists x Fx \to By)$

e.

1. | $\exists x(Fx \to \forall y Gy)$
2. | | $Fj \to \forall y Gy$
3. | | | Fj
4. | | | $\forall y Gy$
5. | | | Gk
6. | | $Fj \to Gk$
7. | | $\forall y(Fj \to Gy)$
8. | | $\exists x \forall y(Fx \to Gy)$
9. | $\exists x \forall y(Fx \to Gy)$

E6.27. The following are not legitimate *ND* derivations. In each case, explain why.

*a.

1. | $Gjy \to Fjy$ P
2. | $\forall z(Gzy \to Fjy)$ 1 $\forall$I

b.

1. | $\exists x \forall y Byx$ P
2. | | $\forall y Byy$ A (g, 1$\exists$E)
3. | | Baa 2 $\forall$E
4. | Baa 1,2-3 $\exists$E

c.

1. | $\exists x Byx$ P
2. | | Byy A (g, 1$\exists$E)
3. | | $\exists y Byy$ 2 $\exists$I
4. | $\exists y Byy$ 1,2-3 $\exists$E

d.

1.	$\forall x\exists yLxy$	P
2.	$\exists yLjy$	1 $\forall$E
3.	$\quad Ljk$	A (g, 2 $\exists$E)
4.	$\quad \forall xLxk$	3 $\forall$I
5.	$\quad \exists y\forall xLxy$	4 $\exists$I
6.	$\exists y\forall xLxy$	2,3-5 $\exists$E

e.

1.	$\forall x(Hx \to Gx)$	P
2.	$\exists xHx$	P
3.	$\quad Hj$	A (g, 2$\exists$E)
4.	$\quad Hj \to Gj$	1 $\forall$E
5.	$\quad Gj$	4,3 $\to$E
6.	Gj	2,3-5 $\exists$E
7.	$\forall xGx$	6 $\forall$I

E6.28. Provide derivations to show each of the following.

a. $\forall xKxx \vdash_{ND} \forall zKzz$

b. $\exists xKxx \vdash_{ND} \exists zKzz$

*c. $\forall x{\sim}Kx, \forall x({\sim}Kx \to {\sim}Sx) \vdash_{ND} \forall x(Hx \vee {\sim}Sx)$

d. $\vdash_{ND} \forall xHf^1x \to \forall xHf^1g^1x$

e. $\forall x\forall y(Gy \to Fx) \vdash_{ND} \forall x(\forall yGy \to Fx)$

*f. $\exists yByyy \vdash_{ND} \exists x\exists y\exists zBxyz$

g. $\forall x[(Hx \wedge {\sim}Kx) \to Ix], \exists y(Hy \wedge Gy), \forall x(Gx \wedge {\sim}Kx) \vdash_{ND} \exists y(Iy \wedge Gy)$

h. $\forall x(Ax \to Bx) \vdash_{ND} \exists zAz \to \exists zBz$

i. $\exists x{\sim}(Cx \vee {\sim}Rx) \vdash_{ND} \exists x{\sim}Cx$

j. $\exists x(Nx \vee Lxx), \forall x{\sim}Nx \vdash_{ND} \exists yLyy$

k. $\forall x\forall y(Fx \to Gy) \vdash_{ND} \forall x(Fx \to \forall yGy)$

l. $\forall x(Fx \to \forall yGy) \vdash_{ND} \forall x\forall y(Fx \to Gy)$

m. $\exists x(Mx \wedge {\sim}Kx), \exists y({\sim}Oy \wedge Wy) \vdash_{ND} \exists x\exists y({\sim}Kx \wedge {\sim}Oy)$

n. $\forall x(Fx \to \exists yGxy) \vdash_{ND} \forall x[Fx \to \exists y(Gxy \vee {\sim}Hxy)]$

o. $\forall x\exists yRxy, \forall x\forall y(Rxy \to Ryx) \vdash_{ND} \forall x\exists y(Rxy \wedge Ryx)$

6.3.3 Strategy

Our strategies remain very much as before. They are modified only to accommodate the parallels between $\wedge$ and $\forall$, and between $\vee$ and $\exists$. I restate the strategies in their expanded form, and give some examples of each. As before, we begin with strategies for reaching a determinate goal.

SG
1. If accessible lines contain explicit contradiction, use $\sim$E to reach goal.
2. Given an accessible formula with main operator $\exists$ or $\vee$, use $\exists$E or $\vee$E to reach goal (watch "screened" variables).
3. If goal is "in" accessible lines (set goals and) attempt to exploit it out.
4. To reach goal with main operator $\star$, use $\star$I (careful with $\vee$ and $\exists$).
5. Try $\sim$E (especially for atomics and formulas with $\vee$ or $\exists$ as main operator).

And we have strategies for reaching a contradiction.

SC
1. Break accessible formulas down into atomics and negated atomics.
2. Given an existential or disjunction in a subderivation for $\sim$E or $\sim$I, go for $\perp$ by $\exists$E or $\vee$E (watch "screened" variables).
3. Set as goal the opposite of some negation (something that cannot itself be broken down). Then apply strategies for a goal to reach it.
4. For some $\mathcal{P}$ such that both $\mathcal{P}$ and $\sim\mathcal{P}$ lead to contradiction: Assume $\mathcal{P}$ ($\sim\mathcal{P}$), obtain the first contradiction, and conclude $\sim\mathcal{P}$ ($\mathcal{P}$); then obtain the second contradiction — this is the one you want.

As before, these are listed in priority order, though the frequency order may be different. If a high priority strategy does not apply, simply fall through to one that does. In each case, you may want to refer back to the corresponding discussion in the sentential case for further discussion and examples.

SG1. *If accessible lines contain explicit contradiction, use $\sim$E to reach goal*. The strategy is unchanged from before. If premises contain an explicit contradiction, we can assume the negation of our goal, bring the contradiction under the assumption, and conclude to the original goal. Since this always works, we want to jump on it whenever it is available. The only thing to add for the quantificational case is that

accessible lines might "contain" a contradiction that is just a short step away buried in quantified expressions. Thus, for example,

(BJ)

1.	$\forall xFx$	P
2.	$\forall y{\sim}Fy$	P
	Gz	

1.	$\forall xFx$	P
2.	$\forall y{\sim}Fy$	P
3.	$\quad {\sim}Gz$	A (g, ~E)
4.	$\quad Fx$	1 ∀E
5.	$\quad {\sim}Fx$	2 ∀E
6.	$\quad \bot$	4,5 ⊥I
7.	Gz	3-6 ~E

Though $\forall xFx$ and $\forall y{\sim}Fy$ are not themselves an explicit contradiction, they lead by ∀E directly to expressions that are. Given the analogy between $\wedge$ and $\forall$, it is as if we had $F \wedge G$ and ${\sim}F \wedge G$ in the premises. In the sentential case, we would not hesitate to go for the goal by ~E. And similarly here.

SG2. *Given an accessible formula with main operator ∃ or ∨, use ∃E or ∨E to reach goal* (watch "screened" variables). What is new for this strategy is the existential quantifier. Motivation is the same as before: With goal $\mathcal{Q}$, and an accessible line with main operator ∃, go for the goal by ∃E. Then you have all the same accessible formulas as before, with the addition of the assumption. So you will (typically) be better off in your attempt to reach $\mathcal{Q}$. We have already emphasized this strategy in introducing the rules. Here is an example.

(BK)

1.	$\exists xFx$	P
2.	$\exists yGy$	P
3.	$\exists zFz \to \forall yFy$	P
4.	$\quad Fj$	A (g, 1∃E)
5.	$\qquad Gk$	A (g, 2∃E)
	$\qquad \exists x(Fx \wedge Gx)$	
	$\quad \exists x(Fx \wedge Gx)$	2, 5-_ ∃E
	$\exists x(Fx \wedge Gx)$	1, 4-_ ∃E

1.	$\exists xFx$	P
2.	$\exists yGy$	P
3.	$\exists zFz \to \forall yFy$	P
4.	$\quad Fj$	A (g, 1∃E)
5.	$\qquad Gk$	A (g, 2∃E)
6.	$\qquad \exists zFz$	4 ∃I
7.	$\qquad \forall yFy$	3,6 →E
8.	$\qquad Fk$	7 ∀E
9.	$\qquad Fk \wedge Gk$	8,5 ∧I
10.	$\qquad \exists x(Fx \wedge Gx)$	9 ∃I
11.	$\quad \exists x(Fx \wedge Gx)$	2, 5-10 ∃E
12.	$\exists x(Fx \wedge Gx)$	1, 4-11 ∃E

The premise at (3) has main operator → and so is not existentially quantified. But the first two premises have main operator ∃. So we set up to reach the goal with two applications of ∃E. It does not matter which we do first, as either way, we end up

with the same accessible formulas to reach the goal at the innermost subderivation. Once we have the subderivations set up, the rest is straightforward.

Given what we have said, it might appear mysterious how one could be anything but better off going directly for a goal by $\exists$E or $\vee$E. But consider the derivations below.

(BL)

1.	$\forall x \exists y Fxy$	P
2.	$\forall x \forall y(Fxy \rightarrow Gxy)$	P
3.	$\exists y Fjy$	1 $\forall$E
4.	$\quad Fjk$	A (g, 3 $\exists$E)
5.	$\quad \forall y(Fjy \rightarrow Gjy)$	2 $\forall$E
6.	$\quad Fjk \rightarrow Gjk$	5 $\forall$E
7.	$\quad Gjk$	6,4 $\rightarrow$E
8.	$\quad \exists y Gjy$	7 $\exists$I
9.	$\quad \forall x \exists y Gxy$	!Mistake!
10.	$\forall x \exists y Gxy$	3,4-9 $\exists$E

(BM)

1.	$\forall x \exists y Fxy$	P
2.	$\forall x \forall y(Fxy \rightarrow Gxy)$	P
3.	$\exists y Fjy$	1 $\forall$E
4.	$\quad Fjk$	A (g, 3 $\exists$E)
5.	$\quad \forall y(Fjy \rightarrow Gjy)$	2 $\forall$E
6.	$\quad Fjk \rightarrow Gjk$	5 $\forall$E
7.	$\quad Gjk$	6,4 $\rightarrow$E
8.	$\quad \exists y Gjy$	7 $\exists$I
9.	$\exists y Gjy$	3,4-8 $\exists$E
10.	$\forall x \exists y Gxy$	9 $\forall$I

In derivation (BL), we isolate the existential on line (3) and go for the goal, $\forall x \exists y Gxy$ by $\exists$E. But something is in fact lost when we set up for the subderivation — the variable j, that was not in any undischarged assumption and therefore available for $\forall$I, gets "screened off" by the assumption and so lost for universal generalization. So at step (9), we are blocked from using (8) and $\forall$I to reach the goal. The problem is solved in (BM) by letting variable j pass into the subderivation and back out, where it is available again for $\forall$I. This requires passing over our second strategy for a goal for at least a step, to set up a new goal $\exists y Gjy$, to which we apply the second strategy in the usual way. Observe that the restriction on $\exists$E blocks a goal in which k is free, but there is no problem about j. This simple case illustrates the sort of context where caution is required in application of SG2.

SG3. *If goal is "in" accessible lines (set goals and) attempt to exploit it out.* This is the same strategy as before. The only thing to add is that we should consider the instances of a universally quantified expression as already "in" the expression (as if it were a big conjunction). Thus, for example,

(BN)

1.	$Ga \rightarrow \forall x Fx$	P
2.	$\forall x Gx$	P
	$\forall x Fx$	
	Fa	_ $\forall$E

1.	$Ga \rightarrow \forall x Fx$	P
2.	$\forall x Gx$	P
3.	Ga	2 $\forall$E
4.	$\forall x Fx$	1,3 $\rightarrow$E
5.	Fa	4 $\forall$E

The original goal Fa is "in" the consequent of (1), $\forall xFx$. So we set $\forall xFx$ as a subgoal. This leads to Ga as another subgoal, and we find this "in" the premise at (2). Very often, the difficult part of a derivation is deciding how to exploit quantifiers to reach a goal. In this case, the choice was trivial. But it is not always so easy.

SG4. *To reach goal with main operator $\star$, use* $\star$I (careful with $\vee$ and $\exists$). As before, this is your "bread-and-butter" strategy. You will come to it over and over. Of new applications, the most automatic is for $\forall$. For a simple case,

(BO)

1.	$\forall xGx$	P
2.	$\forall yFy$	P
	$Fj \wedge Gj$	
	$\forall z(Fz \wedge Gz)$	_ $\forall$I

1.	$\forall xGx$	P
2.	$\forall yFy$	P
3.	Gj	1 $\forall$E
4.	Fj	2 $\forall$E
5.	$Fj \wedge Gj$	4,3 $\wedge$I
6.	$\forall z(Fz \wedge Gz)$	5 $\forall$I

Given a goal with main operator $\forall$, we immediately set up to get it by $\forall$I. This leads to $Fj \wedge Gj$ with the new variable j as a subgoal. After that, completing the derivation is easy. Observe that this strategy does not always work for formulas with main operator $\vee$ and $\exists$.

SG5. *Try* $\sim$E (especially for atomics and formulas with $\vee$ or $\exists$ as main operator). Recall that atomics now include more than just sentence letters. Thus, for example, this rule might have special application for goals of the sort Fab or Gz. And, just as one might have good reason to accept that $\mathscr{P}$ or $\mathscr{Q}$, without having good reason to accept that $\mathscr{P}$, or that $\mathscr{Q}$, so one might have reason to accept that $\exists x\mathscr{P}$ without having to accept that any particular individual is $\mathscr{P}$ — as one might be quite confident that *someone* did it, without evidence sufficient to convict any particular individual. Thus there are contexts where it is possible to derive $\exists x\mathscr{P}$ but not possible to reach it directly by $\exists$I. SG5 has special application in those contexts. Thus, consider the following example.

(BP)

1.	$\sim\forall x Ax$	P
2.	$\sim\exists x{\sim}Ax$	A (c, $\sim$E)
	$\bot$	
	$\exists x{\sim}Ax$	2-_ $\sim$E

1.	$\sim\forall x Ax$	P
2.	$\sim\exists x{\sim}Ax$	A (c, $\sim$E)
3.	$\sim Aj$	A (c, $\sim$E)
4.	$\exists x{\sim}Ax$	3 $\exists$I
5.	$\bot$	4,1 $\bot$I
6.	Aj	3-5 $\sim$E
7.	$\forall x Ax$	6 $\forall$I
8.	$\bot$	7,2 $\bot$I
9.	$\exists x{\sim}Ax$	2-8 $\sim$E

Our initial goal is $\exists x{\sim}Ax$. There is no contradiction in the premises; there is no disjunction or existential in the premises; we do not see the goal in the premises; and attempts to reach the goal by $\exists$I are doomed to fail. So we fall through to SG5, and set up to reach the goal by $\sim$E. As it happens, the contradiction is not easy to get! We can think of the derivation as involving applications of either SC3 or SC4. We take up this sort of case below. For now, the important point is just the setup on the left.

Where strategies for a goal apply in the context of some determinate goal, strategies for a contradiction apply when the goal is just some contradiction — and any contradiction will do. Again, there is nothing fundamentally changed from the sentential case, though we can illustrate some special quantificational applications.

SC1. *Break accessible formulas down into atomics and negated atomics.* This works just as before. The only point to emphasize for the quantificational case is one we made for SG1 above, that relevant atomics may be "contained" in quantified expressions. So going for atomics and negated atomics may include "shaking" quantified expressions to see what falls out. Here is a simple example.

(BQ)

1.	$\sim Fa$	P
2.	$\forall x(Fx \wedge Gx)$	A (c, $\sim$I)
	$\bot$	
	$\sim\forall x(Fx \wedge Gx)$	2-_ $\sim$I

1.	$\sim Fa$	P
2.	$\forall x(Fx \wedge Gx)$	A (c, $\sim$I)
3.	$Fa \wedge Ga$	2 $\forall$E
4.	Fa	3 $\wedge$E
5.	$\bot$	4,1 $\bot$I
6.	$\sim\forall x(Fx \wedge Gx)$	2-5 $\sim$I

Our strategy for the goal is SG4. For an expression with main operator $\sim$, we go for the goal by $\sim$I. We already have $\sim Fa$ toward a contradiction at the level of atomics and negated atomics. And Fa comes from the universally quantified expression by $\forall$E.

SC2. *Given an existential or disjunction in a subderivation for ∼E or ∼I, go for ⊥ by ∃E or ∨E* (watch "screened" variables). Where applications of this strategy were infrequent in the sentential case, they will be much more common now. Motivation is unchanged from SG2: In your attempt to reach a contradiction, you have all the same accessible formulas as before, with the addition of the assumption. So you will (typically) be better off in your attempt to reach a contradiction. Here is an example.

(BR)

1.	$\forall x{\sim}Ax$	P
2.	$\quad \exists xAx$	A (c, ∼I)
	$\quad \bot$	
	${\sim}\exists xAx$	2-_ ∼I

1.	$\forall x{\sim}Ax$	P
2.	$\quad \exists xAx$	A (c, ∼I)
3.	$\quad\quad Aj$	A (c, 2∃E)
	$\quad\quad \bot$	
	$\quad \bot$	2,3-_ ∃E
	${\sim}\exists xAx$	2-_ ∼I

We set up to reach the main goal by ∼I. This gives us an existentially quantified expression at (2), where the goal is a contradiction. SC2 tells us to go for ⊥ by ∃E. Observe that, because the goal is ⊥, the exit strategy is c rather than g. But by application of SC1, this subderivation is easy.

1.	$\forall x{\sim}Ax$	P
2.	$\quad \exists xAx$	A (c, ∼I)
3.	$\quad\quad Aj$	A (c, 2∃E)
4.	$\quad\quad {\sim}Aj$	1 ∀E
5.	$\quad\quad \bot$	3,4 ⊥I
6.	$\quad \bot$	2,3-5 ∃E
7.	${\sim}\exists xAx$	2-6 ∼I

With Aj on line (3) and ${\sim}Aj$ "contained" on line (1), the derivation is easy. But as occurs with the parallel goal-directed strategy, the the contradiction would not even have been possible without the assumption Aj for ∃E.

As can occur with applications of SG2, it is wise to be careful about applications of this strategy when assumptions for ∃E or ∨E "screen off" variables that would otherwise be available for ∀I. Here is a version of the example from before to illustrate the point.

(BS)

1.	$\sim\forall x\exists yGxy$	P
2.	$\forall x\forall y(Fxy \to Gxy)$	P
3.	$\forall x\exists yFxy$	A (c ~I)
4.	$\exists yFjy$	3 ∀E
5.	Fjk	A (c, 4 ∃E)
6.	$\forall y(Fjy \to Gjy)$	2 ∀E
7.	$Fjk \to Gjk$	6 ∀E
8.	Gjk	7,5 →E
9.	$\exists yGjy$	8 ∃I
10.	$\forall x\exists yGxy$	!Mistake!
11.	$\perp$	10,1 ⊥I
12.	$\perp$	4,5-11 ∃E
13.	$\sim\forall x\exists yFxy$	3-12 ~I

(BT)

1.	$\sim\forall x\exists yGxy$	P
2.	$\forall x\forall y(Fxy \to Gxy)$	P
3.	$\forall x\exists yFxy$	A (c ~I)
4.	$\exists yFjy$	3 ∀E
5.	Fjk	A (g, 4 ∃E)
6.	$\forall y(Fjy \to Gjy)$	2 ∀E
7.	$Fjk \to Gjk$	6 ∀E
8.	Gjk	7,5 →E
9.	$\exists yGjy$	8 ∃I
10.	$\exists yGjy$	4,5-9 ∃E
11.	$\forall x\exists yGxy$	10 ∀I
12.	$\perp$	11,1 ⊥I
13.	$\sim\forall x\exists yFxy$	3-12 ~I

In derivation (BS), we isolate the existential on line (4) and set up to go for contradiction by ∃E. But something is in fact lost when we set up for the subderivation — the variable j, that was not in any undischarged assumption and therefore available for ∀I, gets "screened off" by the assumption and so lost for universal generalization. So at step (10), we are blocked from using (9) and ∀I to reach the goal. Again, the problem is solved in (BT) by letting variable j pass into the subderivation and back out, where it is available for ∀I. We do this by letting the goal for ∃E be not ⊥ but rather the formula which results in ⊥, and obtaining ⊥ once we get that formula out. This simple case illustrates the sort of context where caution is required in application of SC2.

SC3. *Set as goal the opposite of some negation (something that cannot itself be broken down); then apply strategies for a goal to reach it.* In principle, this strategy is unchanged from before, though of course there are new applications for quantified expressions. (BT) above includes a case of this. Here is another quick example.

(BU)

1.	$\sim\exists xAx$	P
2.	Aj	A (c, ~I)
	$\perp$	
	$\sim Aj$	2-_ ~I
	$\forall x{\sim}Ax$	_ ∀I

1.	$\sim\exists xAx$	P
2.	Aj	A (c, ~I)
3.	$\exists xAx$	2 ∃I
4.	$\perp$	3,1 ⊥I
5.	$\sim Aj$	2-4 ~I
6.	$\forall x{\sim}Ax$	5 ∀I

Our strategy for the goal is SG4. We plan on reaching $\forall x{\sim}Ax$ by ∀I. So we set $\sim Aj$ as a subgoal. Again the strategy for the goal is SG4, and we set up to get $\sim Aj$

by ∼I. Other than the assumption itself, there are no atomics and negated atomics to be had. There is no existential or disjunction in the scope of the subderivation. But the premise is a negated expression. So we set $\exists x Ax$ as a goal. But this is easy as it comes in one step by ∃I.

SC4. *For some $\mathcal{P}$ such that both $\mathcal{P}$ and $\sim\mathcal{P}$ lead to contradiction: Assume $\mathcal{P}$ ($\sim\mathcal{P}$), obtain the first contradiction, and conclude $\sim\mathcal{P}$ ($\mathcal{P}$); then obtain the second contradiction* — this is the one you want. As in the sentential case, this strategy often coincides with SC3 — in building up to the opposite of something that cannot be broken down, one assumes a $\mathcal{P}$ such that both $\mathcal{P}$ and $\sim\mathcal{P}$ result in contradiction. Corresponding to the pattern with $\vee$, this often happens when some accessible expression is a negated existential. Here is a challenging example.

(BV)

1.	$\forall x(\sim Ax \rightarrow Kx)$	P
2.	$\sim\forall y Ky$	P
3.	$\sim\exists w Aw$	A (c, ∼E)
	$\bot$	
	$\exists w Aw$	3- ∼E

1.	$\forall x(\sim Ax \rightarrow Kx)$	P
2.	$\sim\forall y Ky$	P
3.	$\sim\exists w Aw$	A (c, ∼E)
4.	Aj	A (c, ∼I)
5.	$\exists w Aw$	4 ∃I
6.	$\bot$	5,3 ⊥I
7.	$\sim Aj$	4-6 ∼I
8.	$\sim Aj \rightarrow Kj$	1 ∀E
9.	Kj	8,7 →E
10.	$\forall y Ky$	9 ∀I
11.	$\bot$	10,2 ⊥I
12.	$\exists w Aw$	3-11 ∼E

Once we decide that we cannot get the goal directly be ∃I, the strategy for a goal falls through to SG5. And, as it turns out, both Aj and $\sim Aj$ lead to contradiction. So we assume one and get the contradiction; this gives us the other which leads to contradiction as well. The decision to assume Aj may seem obscure! But it is a common pattern: Given $\sim\exists x\mathcal{P}$, assume an instance, $\mathcal{P}^x_v$ for some variable v, or at least something that will yield $\mathcal{P}^x_v$. Then ∃I gives you $\exists x\mathcal{P}$, and so the first contradiction. So you conclude $\sim\mathcal{P}^x_v$ — and this *outside* the scope of the assumption, where ∀I and the like might apply for v. In effect, you come with an an instance "underneath" the negated existential, where the result is a negation of the instance, which has some chance to give you what you want. For another example of this pattern, see (BP) above.

Notice that such cases can also be understood as driven by applications of SC3. In (BV), we set the opposite of the formula on (2) as goal. This leads to Kj and

then $\sim Aj$ as subgoals. To reach $\sim Aj$, we assume Aj, and get this by building to the opposite of $\sim\exists wAw$. And similarly in (BP).

Again, these strategies are not a cookbook for performing all derivations — doing derivations remains an art. But the strategies will give you a good start, and take you a long way through the exercises that follow, including derivation of the theorems immediately below.

T6.27. $\vdash_{ND} \forall x\mathcal{P} \to \mathcal{P}^x_t$ where term t is free for variable x in formula $\mathcal{P}$

*T6.28. $\mathcal{P} \to \mathcal{Q} \vdash_{ND} \mathcal{P} \to \forall x\mathcal{Q}$ where variable x is not free in formula $\mathcal{P}$

T6.29. $\vdash_{ND} \sim\forall x\mathcal{P} \leftrightarrow \exists x\sim\mathcal{P}$ for any variable x and formula $\mathcal{P}$

T6.30. $\vdash_{ND} \sim\exists x\mathcal{P} \leftrightarrow \forall x\sim\mathcal{P}$ for any variable x and formula $\mathcal{P}$

E6.29. For each of the following, (i) which strategies for a goal apply? and (ii) show the next two steps. If the strategies call for a new subgoal, show the subgoal; if they call for a subderivation, set up the subderivation. In each case, *explain* your response. Hint: each of the strategies for a goal is used at least once.

*a.

1.	$\exists x\exists y(Fxy \wedge Gyx)$	P
	$\exists x\exists yFyx$	

b.

1.	$\forall y[(Hy \wedge Fy) \to Gy]$	P
2.	$\forall zFz \wedge \sim\forall xKxb$	P
	$\forall x(Hx \to Gx)$	

c.

1.	$\forall x[Fx \to \forall y(Gy \to Rxy)]$	P
2.	$\forall x(Hx \to Gx)$	P
3.	$Fa \wedge Hb$	P
	Rab	

d.
1. $\forall x \forall y(Rxy \to \sim Ryx)$ P
2. Raa P

$\exists z \exists y Syz$

e.
1. $\sim\forall x(Fx \vee A)$ P

$\exists x \sim Fx$

E6.30. Each of the following sets up an application of $\sim$I or $\sim$E for SG4 or SG5. Complete the derivations, and *explain* your use of strategies for a contradiction. Hint: Each of the strategies for a contradiction is used at least once.

*a.
1. $\sim\exists x(Fx \wedge Gx)$ P
2. Fj A (g, $\to$I)
3. Gj A (c, $\sim$I)

$\perp$

$\sim Gj$ 3- _ $\sim$I

$Fj \to \sim Gj$ 2- _ $\to$I

$\forall x(Fx \to \sim Gx)$ _ $\forall$I

b.
1. $\forall x(Fx \to \forall y \sim Fy)$ P
2. $\exists x Fx$ A (c, $\sim$I)

$\perp$

$\sim\exists x Fx$ 2- _ $\sim$I

c.
1. $\forall x(Fx \to \forall y Rxy)$ P
2. $\sim Rab$ P
3. Fa A (c, $\sim$I)

$\perp$

$\sim Fa$ 3- _ $\sim$I

d.

1.	$\sim\forall x Fx$	P
2.	$\quad\sim\exists x(\sim Fx \vee A)$	A (c, $\sim$E)
	$\quad\bot$	
	$\exists x(\sim Fx \vee A)$	2-_ $\sim$E

e.

1.	$\quad\exists x(Ax \leftrightarrow \sim Ax)$	A (c, $\sim$I)
	$\quad\bot$	
	$\sim\exists x(Ax \leftrightarrow \sim Ax)$	1-_ $\sim$I

E6.31. Produce derivations to show each of the following. Though no full answers are provided, strategy hints are available for the first problems. If you get the last few on your own, you are doing very well!

*a. $\forall x(\sim Bx \to \sim Wx), \exists x Wx \vdash_{ND} \exists x Bx$

*b. $\forall x\forall y\forall z Gxyz \vdash_{ND} \forall x\forall y\forall z(Hxyz \to Gzyx)$

*c. $\forall x[Ax \to \forall y(\sim Dxy \leftrightarrow Bf^1f^1y)], \forall x(Ax \wedge \sim Bx) \vdash_{ND} \forall x Df^1xf^1x$

*d. $\forall x(Hx \to \forall y Rxyb), \forall x\forall z(Razx \to Sxzz) \vdash_{ND} Ha \to \exists x Sxcc$

*e. $\sim\forall x(Fx \wedge Abx) \leftrightarrow \sim\forall x Kx, \forall y[\exists x\sim(Fx \wedge Abx) \wedge Ryy] \vdash_{ND} \sim\forall x Kx$

*f. $\exists x(Jxa \wedge Cb), \exists x(Sx \wedge Hxx), \forall x[(Cb \wedge Sx) \to \sim Ax] \vdash_{ND} \exists z(\sim Az \wedge Hzz)$

*g. $\forall x\forall y(Dxy \to Cxy), \forall x\exists y Dxy, \forall x\forall y(Cyx \to Dxy) \vdash_{ND} \exists x\exists y(Cxy \wedge Cyx)$

*h. $\forall x\forall y[(Ry \vee Dx) \to \sim Ky], \forall x\exists y(Ax \to \sim Ky), \exists x(Ax \vee Rx) \vdash_{ND} \exists x\sim Kx$

*i. $\forall y(My \to Ay), \exists x\exists y[(Bx \wedge Mx) \wedge (Ry \wedge Syx)], \exists x Ax \to \forall y\forall z(Syz \to Ay)$ $\vdash_{ND} \exists x(Rx \wedge Ax)$

*j. $\forall x\forall y[(Hby \wedge Hxb) \to Hxy], \forall z(Bz \to Hbz), \exists x(Bx \wedge Hxb)$ $\vdash_{ND} \exists z[Bz \wedge \forall y(By \to Hzy)]$

*k. $\forall x((Fx \wedge \sim Kx) \to \exists y[(Fy \wedge Hyx) \wedge \sim Ky])$, $\forall x[(Fx \wedge \forall y[(Fy \wedge Hyx) \to Ky]) \to Kx] \to Ma \vdash_{ND} Ma$

*l. $\forall x\forall y[(Gx \wedge Gy) \to (Hxy \to Hyx)], \forall x\forall y\forall z([(Gx \wedge Gy) \wedge Gz] \to$ $[(Hxy \wedge Hyz) \to Hxz]) \vdash_{ND} \forall w([Gw \wedge \exists z(Gz \wedge Hwz)] \to Hww)$

*m. $\forall x\forall y[(Ax \wedge By) \to Cxy], \exists y[Ey \wedge \forall w(Hw \to Cyw)], \forall x\forall y\forall z[(Cxy \wedge$ $Cyz) \to Cxz], \forall w(Ew \to Bw) \vdash_{ND} \forall z\forall w[(Az \wedge Hw) \to Czw]$

*n. $\forall x \exists y \forall z(Axyz \vee Bzyx), \sim\exists x \exists y \exists z Bzyx \vdash_{ND} \forall x \exists y \forall z Axyz$

*o. $A \rightarrow \exists x Fx \vdash_{ND} \exists x(A \rightarrow Fx)$

*p. $\forall x Fx \rightarrow A \vdash_{ND} \exists x(Fx \rightarrow A)$

q. $\forall x(Fx \rightarrow Gx), \forall x \forall y(Rxy \rightarrow Syx), \forall x \forall y(Sxy \rightarrow Syx)$
$\vdash_{ND} \forall x[\exists y(Fx \wedge Rxy) \rightarrow \exists y(Gx \wedge Sxy)]$

r. $\exists y \forall x Rxy, \forall x(Fx \rightarrow \exists y Syx), \forall x \forall y(Rxy \rightarrow \sim Sxy) \vdash_{ND} \exists x \sim Fx$

s. $\exists x \forall y[(Fx \vee Gy) \rightarrow \forall z(Hxy \rightarrow Hyz)], \exists z \forall x \sim Hxz \vdash_{ND} \exists y \forall x(Fy \rightarrow \sim Hyx)$

t. $\forall x \forall y[\exists z Hyz \rightarrow Hxy] \vdash_{ND} \exists x \exists y Hxy \rightarrow \forall x \forall y Hxy$

u. $\exists x(Fx \wedge \forall y[(Gy \wedge Hy) \rightarrow \sim Sxy]), \forall x \forall y([(Fx \wedge Gy) \wedge Jy] \rightarrow \sim Sxy),$
$\forall x \forall y([(Fx \wedge Gy) \wedge Rxy] \rightarrow Sxy), \exists x(Gx \wedge (Jx \vee Hx))$
$\vdash_{ND} \exists x \exists y((Fx \wedge Gy) \wedge \sim Rxy)$

v. $\vdash_{ND} \exists x \forall y(Fx \rightarrow Fy)$

w. $\vdash_{ND} \exists x(\exists y Fy \rightarrow Fx)$

x. $\exists x \forall y[\exists z(Fzy \rightarrow \exists w Fyw) \rightarrow Fxy] \vdash_{ND} \exists x Fxx$

y. $\vdash_{ND} \forall x \exists y \forall z[\exists w Txyw \rightarrow \exists w Txzw]$

z. $\vdash_{ND} \forall x \exists y(Fx \vee Gy) \rightarrow \exists y \forall x(Fx \vee Gy)$

*E6.32. Produce derivations to demonstrate each of T6.27 - T6.30, explaining for each application how quantifier restrictions are met. Hint: You might try working test versions where $\mathcal{P}$ and $\mathcal{Q}$ are atomics Px and Qx; then you can think about the general case.

6.3.4 =I and =E

We complete the system *ND* with I- and E- rules for equality. Strictly, $=$ is not an operator at all; it is a two-place relation symbol. However, because its interpretation is standardized across all interpretations, it is possible to introduce rules for its behavior. The =I rule is particularly simple. At any stage in a derivation, for any term t, one may write down $t = t$ with justification =I.

=I $\quad | \; t = t \qquad$ =I

Strictly, without any inputs, this is an *axiom* of the sort we encountered in chapter 3. It is a formula which may be asserted at any stage in a derivation. Its motivation should be clear. Since for any m in the universe U, $\langle \mathsf{m}, \mathsf{m} \rangle$ is in the interpretation of $=$, $t = t$ is sure to be satisfied, no matter what the assignment to t might be. Thus, in $\mathcal{L}_q$, $a = a$, $x = x$, and $f^2az = f^2az$ are formulas that might be justified by =I.

=E is more interesting and, in practice, more useful. Say an arbitrary term is *free* in a formula iff every variable in it is free. Automatically, then, any term without variables is free in any formula. And say $\mathcal{P}^{t}/_{s}$ is $\mathcal{P}$ where some, but not necessarily all, free instances of term t may be replaced by term s. Then, given an accessible formula $\mathcal{P}$ on line a and the atomic formula $t = s$ or $s = t$ on accessible line b, one may move to $\mathcal{P}^{t}/_{s}$, where s is free for all the replaced instances of t in $\mathcal{P}$, with justification *a,b* =E.

=E

a.	$\mathcal{P}$		a.	$\mathcal{P}$		provided that term s is free for all the replaced instances of term t in formula $\mathcal{P}$
b.	$t = s$		b.	$s = t$		
	$\mathcal{P}^{t}/_{s}$	a,b =E		$\mathcal{P}^{t}/_{s}$	a,b =E	

If the assignment to some terms is the same, this rule lets us replace free instances of the one term by the other in any formula. Again, the motivation should be clear. On trees, the only thing that matters about a term is the thing to which it refers. So if $\mathcal{P}$ with term t is satisfied, and the assignment to t is the same as the assignment to s, then $\mathcal{P}$ with s in place of t should be satisfied as well. When a term is not free, it is not the assignment to the term that is doing the work, but rather the way it is bound. So we restrict ourselves to contexts where it is just the assignment that matters!

Because we need not replace all free instances of one term with the other, this rule has some special applications that are worth noticing. Consider the formulas $Raba$ and $a = b$. The following lists all the formulas that could be derived from them in one step by =E.

(BW)

1.	$Raba$	P
2.	$a = b$	P
3.	$Rbba$	1,2 =E
4.	$Rabb$	1,2 =E
5.	$Rbbb$	1,2 =E
6.	$Raaa$	1,2 =E
7.	$a = a$	2,2 =E
8.	$b = b$	2,2 =E

(3) and (4) replace one instance of a with b. (5) replaces both instances of a with b. (6) replaces the instance of b with a. We could reach, say, $Raab$, but this would

ND Quick Reference (Quantificational)

∀E (*universal exploit*)

$$\begin{array}{r|ll} \text{a.} & \forall x\mathscr{P} & \\ & & \\ & \mathscr{P}^x_t & \text{a } \forall\text{E} \end{array}$$

∃I (*existential intro*)

$$\begin{array}{r|ll} \text{a.} & \mathscr{P}^x_t & \\ & & \\ & \exists x\mathscr{P} & a\ \exists\text{I} \end{array}$$

provided t is free for x in $\mathscr{P}$

∀I (*universal intro*)

$$\begin{array}{r|ll} \text{a.} & \mathscr{P}^x_v & \\ & & \\ & \forall x\mathscr{P} & a\ \forall\text{I} \end{array}$$

∃E (*existential exploit*)

$$\begin{array}{r|l|ll} \text{a.} & \multicolumn{2}{l}{\exists x\mathscr{P}} & \\ \text{b.} & & \mathscr{P}^x_v & \text{A } (g, a\exists\text{E}) \\ & & & \\ \text{c.} & & \mathscr{Q} & \\ & \multicolumn{2}{l}{\mathscr{Q}} & \text{a,b-c } \exists\text{E} \end{array}$$

provided (i) v is free for x in $\mathscr{P}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\forall x\mathscr{P}$ / in $\exists x\mathscr{P}$ or in $\mathscr{Q}$

=I (*equality intro*)

$$\begin{array}{|ll} & \\ t = t & =\text{I} \end{array}$$

=E (*equality exploit*)

$$\begin{array}{r|l} \text{a} & \mathscr{P} \\ \text{b} & t = s \\ & \\ & \mathscr{P}^{t}/_{s} \end{array} \qquad \begin{array}{|ll} \mathscr{P} & \\ s = t & \\ & \\ \mathscr{P}^{t}/_{s} & \text{a,b =E} \end{array}$$

provided that term s is free for all the replaced instances of term t in formula $\mathscr{P}$

require another step — which we could take from any of (4), (5) or (6). You should be clear about why this is so. (7) and (8) are different. We have a formula $a = b$, and an equality $a = b$. In (7) we use the equality to replace one instance of b in the formula with a. In (8) we use the equality to replace one instance of a in the formula with b. Of course (7) and (8) might equally have been derived by =I. Notice also that =E is not restricted to atomic formulas, or to simple terms. Thus, for example,

(BX)

$$\begin{array}{r|ll} 1. & \forall y(Rax \wedge Kxy) & \text{P} \\ 2. & x = f^3azx & \text{P} \\ \hline 3. & \forall y(Raf^3azx \wedge Kxy) & 1,2 = \text{E} \\ 4. & \forall y(Rax \wedge Kf^3azxy) & 1,2 = \text{E} \\ 5. & \forall y(Raf^3azx \wedge Kf^3azxy) & 1,2 = \text{E} \end{array}$$

lists the steps that are legitimate applications of =E to (1) and (2). What we could not do is use, $x = f^3azy$ to with (1) to reach say, $\forall y(Raf^3azy \wedge Kxy)$, since f^3azy is not free for any instance of x in $\forall y(Rax \wedge Kxy)$. And of course, we could not replace any instances of y in $\forall y(Rax \wedge Kxy)$ since none of them are free.

There is not much new to say about strategy, except that you should include =E among the stock of rules you use to identify what is "contained" in the premises. It may be that a goal is contained in the premises, when terms only need to be switched by some equality. Thus, for goal Fa, with Fb explicitly in the premises, it might

be worth setting $a = b$ as a subgoal, with the intent using the equality to switch the terms.

Preliminary Results. Rather than dwell on strategy as such, let us consider a few substantive applications. First, you should find derivation of the following theorems straightforward. Thus, for example, T6.31 and T6.34 take just one step. The first three may remind you of axioms from chapter 3. The others represent important features of equality.

T6.31. $\vdash_{ND} \mathscr{x} = \mathscr{x}$

*T6.32. $\vdash_{ND} (\mathscr{x}_i = \mathscr{y}) \to (\mathscr{h}^n \mathscr{x}_1 \ldots \mathscr{x}_i \ldots \mathscr{x}_n = \mathscr{h}^n \mathscr{x}_1 \ldots \mathscr{y} \ldots \mathscr{x}_n)$

T6.33. $\vdash_{ND} (\mathscr{x}_i = \mathscr{y}) \to (\mathscr{R}^n \mathscr{x}_1 \ldots \mathscr{x}_i \ldots \mathscr{x}_n \to \mathscr{R}^n \mathscr{x}_1 \ldots \mathscr{y} \ldots \mathscr{x}_n)$

T6.34. $\vdash_{ND} \mathscr{t} = \mathscr{t}$ *reflexivity* of equality

T6.35. $\vdash_{ND} (\mathscr{t} = \mathscr{s}) \to (\mathscr{s} = \mathscr{t})$ *symmetry* of equality

T6.36. $\vdash_{ND} (\mathscr{r} = \mathscr{s}) \to [(\mathscr{s} = \mathscr{t}) \to (\mathscr{r} = \mathscr{t})]$ *transitivity* of equality

For a more substantive case, suppose we want to show that the following argument is valid in *ND*.

(BY)

$\exists x[(Dx \wedge \forall y(Dy \to x = y)) \wedge Bx]$	The dog is barking
$\exists x(Dx \wedge Cx)$	Some dog is chasing a cat
$\exists x[Dx \wedge (Bx \wedge Cx)]$	Some dog is barking and chasing a cat

Using the methods of chapter 5, this might translate something like the argument on the right. We set out to do the derivation in the usual way.

1.	$\exists x[(Dx \wedge \forall y(Dy \to x = y)) \wedge Bx]$	P
2.	$\exists x(Dx \wedge Cx)$	P
3.	$\quad (Dj \wedge \forall y(Dy \to j = y)) \wedge Bj$	A (g, 1∃E)
4.	$\quad\quad Dk \wedge Ck$	A (g, 2∃E)
	$\quad\quad Dj \wedge (Bj \wedge Cj)$	
	$\quad\quad \exists x[Dx \wedge (Bx \wedge Cx)]$	_ ∃I
	$\quad \exists x[Dx \wedge (Bx \wedge Cx)]$	2,4-_ ∃E
	$\exists x[Dx \wedge (Bx \wedge Cx)]$	1,3-_ ∃E

Given two existentials in the premises, we set up to get the goal by two applications of $\exists$E. And we can get the conclusion from $Dj \wedge (Bj \wedge Cj)$ by $\exists$I. Dj and Bj are easy to get from (3). But we do not have Cj. What we have is rather Ck. The existentials in the assumptions are instantiated to different (new) variables — and they *must* be so instantiated if we are to meet the constraints on $\exists$E. From $\exists x\mathcal{P}$ and $\exists x\mathcal{Q}$ it does not follow that any one thing is both $\mathcal{P}$ and $\mathcal{Q}$. In this case, however, we are given that there is just one dog. And we can use this to force an equivalence between j and k. Then we get the result by =E.

1.	$\exists x[(Dx \wedge \forall y(Dy \to x = y)) \wedge Bx]$	P
2.	$\exists x(Dx \wedge Cx)$	P
3.	$\quad (Dj \wedge \forall y(Dy \to j = y)) \wedge Bj$	A (g, 1$\exists$E)
4.	$\quad\quad Dk \wedge Ck$	A (g, 2$\exists$E)
5.	$\quad\quad Bj$	3 $\wedge$E
6.	$\quad\quad Dj \wedge \forall y(Dy \to j = y)$	3 $\wedge$E
7.	$\quad\quad Dj$	6 $\wedge$E
8.	$\quad\quad \forall y(Dy \to j = y)$	6 $\wedge$E
9.	$\quad\quad Dk \to j = k$	8 $\forall$E
10.	$\quad\quad Dk$	4 $\wedge$E
11.	$\quad\quad j = k$	9,10 $\to$E
12.	$\quad\quad Ck$	4 $\wedge$E
13.	$\quad\quad Cj$	12,11 =E
14.	$\quad\quad Bj \wedge Cj$	5,13 $\wedge$I
15.	$\quad\quad Dj \wedge (Bj \wedge Cj)$	7,14 $\wedge$I
16.	$\quad\quad \exists x[Dx \wedge (Bx \wedge Cx)]$	15 $\exists$I
17.	$\quad \exists x[Dx \wedge (Bx \wedge Cx)]$	2,4-16 $\exists$E
18.	$\exists x[Dx \wedge (Bx \wedge Cx)]$	1,3-17 $\exists$E

Though there are a few steps, the work to get it done is simple. This is a very common pattern: Arbitrary individuals are introduced as if they were distinct. But uniqueness clauses let us establish an identity between them. Given this, facts about the one transfer to the other by =E.

At this stage, it would be appropriate to take on E6.33 and E6.34.

Robinson Arithmetic, Q. A very important application, already encountered in chapter 3, is to mathematics. For this, $\mathcal{L}_{NT}$ is like $\mathcal{L}_{NT}^{<}$ in section 2.2.5 on p. 63 but without $<$. There are the constant symbol $\emptyset$, the function symbols S, $+$ and $\times$, and the relation symbol $=$. Let $s \leq t$ abbreviate $\exists v(v + s = t)$, and $s < t$ abbreviate $\exists v(Sv + s = t)$ where v is a variable that does not appear in s or t. We shall also require a species of *bounded* quantifiers. So, $(\forall x \leq t)\mathcal{P}$ abbreviates

$\forall x(x \leq t \to \mathcal{P})$ and $(\exists x \leq t)\mathcal{P}$ abbreviates $\exists x(x \leq t \land \mathcal{P})$, and similarly for $(\forall x < t)\mathcal{P}$ and $(\exists x < t)\mathcal{P}$, where x does not occur in t.

Observe that simple derived introduction and exploitation rules are possible for the bounded quantifiers. So, for example,

(∀E)

$$\begin{array}{l|l}
\text{a.} & (\forall x < t)\mathcal{P} \\
\text{b.} & s < t \\
 & \\
 & \mathcal{P}^x_s
\end{array}$$

(∃I)

$$\begin{array}{l|l}
\text{a.} & \mathcal{P}^x_s \\
\text{b.} & s < t \\
 & \\
 & (\exists x < t)\mathcal{P}
\end{array}$$

provided s is free for x in $\mathcal{P}$

(∀I)

$$\begin{array}{l|l}
\text{a.} & \begin{array}{|l} \underline{v < t} \\ \\ \mathcal{P}^x_v \end{array} \\
 & (\forall x < t)\mathcal{P}
\end{array}$$

(∃E)

$$\begin{array}{l|l}
\text{a.} & (\exists x < t)\mathcal{P} \\
\text{b.} & \begin{array}{|l} \mathcal{P}^x_v \end{array} \\
\text{c.} & \begin{array}{|l} \underline{v < t} \\ \\ \mathcal{Q} \end{array} \\
 & \mathcal{Q}
\end{array}$$

provided v is free for x in $\mathcal{P}$, not free in any undischarged assumption and not free in the quantified expression or $\mathcal{Q}$

So, for example, for (∀E), unabbreviation and then ∀E with →E give the desired result. The other cases are just as easy, and left as an exercise.

Officially, formulas of $\mathcal{L}_{\text{NT}}$ may be treated as uninterpreted. It is natural, however, to think of them with their usual meanings, with ∅ for zero, S the successor function, $+$ the addition function, $\times$ the multiplication function, and $=$ the equality relation. But, again, we do not need to think about that for now.

We will say that a formula $\mathcal{P}$ is an *ND theorem of Robinson Arithmetic* just in case $\mathcal{P}$ follows in *ND* given as premises the following axioms for Robinson Arithmetic.[5] These axioms are presented as formulas with free variables. But given ∀I and ∀E, they are equivalent to universally quantified forms — and we might as well have stated the axioms as universally quantified sentences.

Q
1. $\sim(Sx = \emptyset)$
2. $(Sx = Sy) \to (x = y)$
3. $(x + \emptyset) = x$
4. $(x + Sy) = S(x + y)$
5. $(x \times \emptyset) = \emptyset$
6. $(x \times Sy) = [(x \times y) + x]$
7. $\sim(x = \emptyset) \to \exists y(x = Sy)$

[5]After R. Robinson, "An Essentially Undecidable Axiom System." Again (p. 88n2) observe that 'theorem' is context-relative. A theorem of Robinson arithmetic which results only givien Q1 - Q7 is not a theorem of ND just because it takes some of Q1 - Q7 for its derivation.

$\mathcal{L}_{\text{NT}}$ reference

Vocabulary:

constant: $\emptyset$

one-place function symbol: S

two-place function symbols: $+$, $\times$

relation symbol: $=$

Abbreviations:

$s \leq t$ abbreviates $\exists v(v + s = t)$

$s < t$ abbreviates $\exists v(Sv + s = t)$

— where v does not appear in s or t

$(\forall x \leq t)\mathcal{P}$ abbreviates $\forall x(x \leq t \rightarrow \mathcal{P})$

$(\forall x < t)\mathcal{P}$ abbreviates $\forall x(x < t \rightarrow \mathcal{P})$

$(\exists x \leq t)\mathcal{P}$ abbreviates $\exists x(x \leq t \wedge \mathcal{P})$

$(\exists x < t)\mathcal{P}$ abbreviates $\exists x(x < t \wedge \mathcal{P})$

— where x does not appear in t

In *ND*, the bounded quantifiers have natural derived introduction and exploitation rules ($\forall$E), ($\forall$I), ($\exists$E), ($\exists$I) along with a bounded quantifier negation BQN. In addition, on the standard interpretation for number theory there are derived semantic conditions for the inequalities T12.5 and for the bounded quantifiers T12.6 and T12.7.

In the ordinary case we suppress mention of Q1 - Q7 as premises, and simply write Q $\vdash_{ND}$ $\mathcal{P}$ to indicate that $\mathcal{P}$ is an *ND* theorem of Robinson arithmetic — that there is an *ND* derivation of $\mathcal{P}$ which may include appeal to any of Q1 - Q7.

The axioms set up a basic version of arithmetic on the non-negative integers. Intuitively, $\emptyset$ is not the successor of any non-negative integer (Q1); if the successor of x is the same as the successor of y, then x is y (Q2); x plus $\emptyset$ is equal to x (Q3); x plus one more than y is equal to one more than x plus y (Q4); x times $\emptyset$ is equal to $\emptyset$ (Q5); x times one more than y is equal to x times y plus x (Q6); and any number other than $\emptyset$ is a successor (Q7).

If some $\mathcal{P}$ is derived directly from some of Q1 - Q7 then it is trivially an *ND* theorem of Robinson Arithmetic. But if the members of a set Γ are *ND* theorems of Robinson Arithmetic, and $\Gamma \vdash_{ND} \mathcal{P}$, then $\mathcal{P}$ is an *ND* theorem of Robinson Arithmetic as well — for any derivation of $\mathcal{P}$ from some theorems might be extended into

one which derives the theorems, and then goes on from there to obtain $\mathcal{P}$. In the ordinary case, then, we *build* to increasingly complex results: having once demonstrated a theorem by a derivation, we feel free simply to *cite* it as a premise in the next derivation. So the collection of formulas we count as premises increases from one derivation to the next.

Though the application to arithmetic is interesting, there is in principle *nothing different* about derivations for Q from ones we have done before: We are moving from premises to a goal. As we make progress, however, there will be an increasing number of premises available, and it may be relatively challenging to recognize *which* premises are relevant to a given goal.

Let us start with some simple generalizations of Q1 - Q7. As they are stated, Q1 - Q7 are formulas involving *variables*. But they permit derivation of corresponding principles for arbitrary terms s and t. The derivations all follow the same $\forall$I, $\forall$E pattern.

T6.37. Q $\vdash_{ND} \sim(St = \emptyset)$

1.	$\sim(Sx = \emptyset)$	Q1
2.	$\forall x \sim(Sx = \emptyset)$	1 $\forall$I
3.	$\sim(St = \emptyset)$	2 $\forall$E

Observe that since $\sim(Sx = \emptyset)$ has no quantifiers, term t is sure to be free for x in $\sim(Sx = \emptyset)$. So there is no problem about the restriction on $\forall$E. And since t is any term, substituting $\emptyset$ and $(S\emptyset + y)$ and the like for for t, we have that $\sim(S\emptyset = \emptyset)$, $\sim(S(S\emptyset + y) = \emptyset)$ and the like are all instances of T6.37. The next theorems are similar.

T6.38. Q $\vdash_{ND} (St = Ss) \to (t = s)$

1.	$(Sx = Sy) \to (x = y)$	Q2
2.	$\forall u[(Su = Sy) \to (u = y)]$	1 $\forall$I
3.	$\forall v \forall u[(Su = Sv) \to (u = v)]$	2 $\forall$I
4.	$\forall u[(Su = Ss) \to (u = s)]$	3 $\forall$E
5.	$(St = Ss) \to (t = s)$	4 $\forall$E

Observe that for (4) it is important that term s not include any variable u. Thus for this derivation we simply choose u so that it is not a variable in s.

*T6.39. Q $\vdash_{ND} (t + \emptyset) = t$

T6.40. Q $\vdash_{ND}$ $(t + S s) = S(t + s)$

T6.41. Q $\vdash_{ND}$ $(t \times \emptyset) = \emptyset$

T6.42. Q $\vdash_{ND}$ $(t \times S s) = ((t \times s) + t)$

T6.43. Q $\vdash_{ND}$ $\sim(t = \emptyset) \rightarrow \exists y(t = Sy)$
where variable y does not appear in t

Given these results, we are ready for some that are more interesting. Let us show that $1 + 1 = 2$. That is, that $S\emptyset + S\emptyset = SS\emptyset$.

T6.44. Q $\vdash_{ND}$ $S\emptyset + S\emptyset = SS\emptyset$

1.	$(S\emptyset + S\emptyset) = S(S\emptyset + \emptyset)$	T6.40
2.	$(S\emptyset + \emptyset) = S\emptyset$	T6.39
3.	$(S\emptyset + S\emptyset) = SS\emptyset$	1,2 =E

Given the premises, this derivation is simple. Given that $(S\emptyset + \emptyset) = S\emptyset$ from (2), we can replace $S\emptyset + \emptyset$ with $S\emptyset$ by =E. This is just what we do, substituting into the first premise. The first premise is an instance of T6.40 that has $S\emptyset$ for t, and $\emptyset$ for s. (2) is an instance of T6.39 with $S\emptyset$ for t. Be sure you understand each step.

Observe the way Q3 and Q4 work together: Q3 (T6.39) gives the sum of any term with zero; and given the sum of a term with any number, Q4 (T6.40) gives the sum of that term and one more than it. So we can calculate the sum of a term and zero from T6.39, and then with T6.40 get the sum of it and one, then it and two, and so forth. So, for example, Q $\vdash_{ND}$ $SS\emptyset + SSS\emptyset = SSSSS\emptyset$. From T6.40, 2 + 3 depends on 2 + 2; but then 2 + 2 depends on 2 + 1; 2 + 1 on 2 + 0; and we get the latter directly. So we start with T6.39 and T6.40.

(BZ)

1.	$(SS\emptyset + \emptyset) = SS\emptyset$	T6.39
2.	$(SS\emptyset + S\emptyset) = S(SS\emptyset + \emptyset)$	T6.40
3.	$(SS\emptyset + S\emptyset) = SSS\emptyset$	1,2 =E

We use (1) to put the known value of $SS\emptyset + \emptyset$ in to the right side of (2). But now the value of $SS\emptyset + S\emptyset$ is known, and we can use T6.40 again.

(CA)

$$\begin{array}{rll}
1. & (SS\emptyset + \emptyset) = SS\emptyset & \text{T6.39} \\
2. & (SS\emptyset + S\emptyset) = S(SS\emptyset + \emptyset) & \text{T6.40} \\
3. & (SS\emptyset + S\emptyset) = SSS\emptyset & \text{1,2 =E} \\
4. & (SS\emptyset + SS\emptyset) = S(SS\emptyset + S\emptyset) & \text{T6.40} \\
5. & (SS\emptyset + SS\emptyset) = SSSS\emptyset & \text{3,4 =E}
\end{array}$$

This time, we use 3 to put the known value of $SS\emptyset + S\emptyset$ into the right side of (4). And we can use T6.40 again to get the final result. Since we are in *ND*, we sort the premises to the top to get,

(CB)

$$\begin{array}{rll}
1. & (SS\emptyset + \emptyset) = SS\emptyset & \text{T6.39} \\
2. & (SS\emptyset + S\emptyset) = S(SS\emptyset + \emptyset) & \text{T6.40} \\
3. & (SS\emptyset + SS\emptyset) = S(SS\emptyset + S\emptyset) & \text{T6.40} \\
4. & \underline{(SS\emptyset + SSS\emptyset) = S(SS\emptyset + SS\emptyset)} & \text{T6.40} \\
5. & (SS\emptyset + S\emptyset) = SSS\emptyset & \text{1,2 =E} \\
6. & (SS\emptyset + SS\emptyset) = SSSS\emptyset & \text{3,5 =E} \\
7. & (SS\emptyset + SSS\emptyset) = SSSSS\emptyset & \text{4,6 =E}
\end{array}$$

Again, the left term $SS\emptyset$ is given from T6.39; we use multiple applications of T6.40 to increase the next term to $SSS\emptyset$ for the final result. And similarly for multiplication: Q5 (T6.41) gives the product of any term with zero; and given the product of a term with any number, Q6 (T6.42) gives the product of that term and one more than it. So we can calculate the product of a term and zero from T6.41, and then with T6.42 get the product of it and one, it and two, and so forth. Here is a general result of the same type.

T6.45. Q $\vdash_{ND}$ $t + S\emptyset = St$

Hint: You can do this in three lines.

Of course, we may manipulate other operators in the usual way.

(CC)

$$\begin{array}{rll}
1. & \underline{(j + Sk) = S(j + k)} & \text{T6.40} \\
2. & \mid\ \underline{\exists y(j + y = S\emptyset)} & \text{A } (g, \rightarrow\text{I}) \\
3. & \mid\ \mid\ \underline{j + k = S\emptyset} & \text{A } (g, 2\exists\text{E}) \\
4. & \mid\ \mid\ j + Sk = SS\emptyset & \text{1,3 = E} \\
5. & \mid\ \mid\ \exists y(j + y = SS\emptyset) & \text{4 }\exists\text{I} \\
6. & \mid\ \exists y(j + y = SS\emptyset) & \text{2,3-5 }\exists\text{E} \\
7. & \exists y(j + y = S\emptyset) \rightarrow \exists y(j + y = SS\emptyset) & \text{2-6 }\rightarrow\text{I} \\
8. & \forall x[\exists y(x + y = S\emptyset) \rightarrow \exists y(x + y = SS\emptyset)] & \text{7 }\forall\text{I}
\end{array}$$

The basic setup for $\forall$I, $\rightarrow$I and $\exists$E is by now routine. The real work is where we use (1) and (3) to obtain $j + Sk = SS\emptyset$. Above, we have used T6.40 with application

to "closed" terms without free variables, built up from $\emptyset$. But nothing stops this application of the theorem in a generic form. Here are a couple of theorems that will be of interest later.

T6.46. $\mathrm{Q} \vdash_{ND} \forall x(x \leq \emptyset \rightarrow x = \emptyset)$

Hints: Be sure you are clear about what is being asked for; at some stage, you will need to unpack the abbreviation. Do not forget that you can appeal to T6.37 and T6.43.

T6.47. $\mathrm{Q} \vdash_{ND} \forall x{\sim}(x < \emptyset)$

Hint: This reduces to a difficult application of SC4. From $\exists v(Sv + j = \emptyset)$, and using T6.43, assume $j \neq \emptyset$ to obtain a first contradiction; and you will be able to obtain contradiction from $j = \emptyset$ as well. You will need a couple applications of SC2 to extract contradictions from applications of $\exists$E.

With this much, you should be able to work E6.35 now.

Robinson Arithmetic is interesting. Its axioms are sufficient to prove arbitrary facts about particular numbers. Its language and derivation system are just strong enough to support Gödel's incompleteness result, on which it is not possible for a "nicely specified" theory including a sufficient amount of arithmetic to have as consequences $\mathcal{P}$ or ${\sim}\mathcal{P}$ for every $\mathcal{P}$ (Part IV). But we do not need Gödel's result to see that Robinson Arithmetic is incomplete: It turns out that many true generalizations are not provable in Robinson Arithmetic. So, for example, neither $\forall x \forall y[(x \times y) = (y \times x)]$, nor its negation is provable.[6] So Robinson Arithmetic is a particularly weak theory.

Peano Arithmetic. Though Robinson Arithmetic leaves even standard results like commutation for multiplication unproven, it is possible to strengthen the derivation system to obtain such results. Thus such standard generalizations are provable in Peano Arithmetic.[7] This is the system we encountered in chapter 3, but now with ND — so that when $\mathcal{P}$ is derived from the axioms it is an *ND theorem of Peano Arithmetic*. For this, let PA1 - PA6 be the same as Q1 - Q6. Replace Q7 as follows. For any formula $\mathcal{P}$,

---

[6] A semantic demonstration of this negative result is left as an exercise for chapter 7. But we already understand the basic idea from chapter 4: To show that a conclusion does not follow, produce an interpretation on which the axioms are true, but the conclusion is not. The connection between derivations and the semantic results must wait for chapter 10.

[7] After the work of R. Dedekind and G. Peano. For historical discussion, see Wang, "The Axiomatization of Arithmetic."

PA7 $[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})] \to \forall x\mathcal{P}$

is an axiom. If a formula $\mathcal{P}$ applies to $\emptyset$, and for any x, if $\mathcal{P}$ applies to x, then it also applies to Sx, then $\mathcal{P}$ applies to every x. This schema represents the *principle of mathematical induction*. We will have much more to say about the principle of mathematical induction in Part II. For now, it is enough merely to *recognize* its instances. Thus, for example, if $\mathcal{P}$ is $\sim(x = Sx)$, then $\mathcal{P}^x_\emptyset$ is $\sim(\emptyset = S\emptyset)$, and $\mathcal{P}^x_{Sx}$ is $\sim(Sx = SSx)$. So,

$$[\sim(\emptyset = S\emptyset) \wedge \forall x(\sim(x = Sx) \to \sim(Sx = SSx))] \to \forall x\sim(x = Sx)$$

is an instance of the scheme. You should see why this is so.

It will be convenient to have the principle of mathematical induction in a rule form. Given $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$ on accessible lines a and b, one may move to $\forall x\mathcal{P}$ with justification *a,b* IN.

IN

a.	$\mathcal{P}^x_\emptyset$	
b.	$\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$	
	$\forall x\mathcal{P}$	a,b IN

1.	$\mathcal{P}^x_\emptyset$	P
2.	$\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$	P
3.	$[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})] \to \forall x\mathcal{P}$	PA7
4.	$\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$	1,2 $\wedge$I
5.	$\forall x\mathcal{P}$	3,4 $\to$E

The rule is justified from PA7 by reasoning as on the right. That is, given $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$ on accessible lines, one can always conjoin them, then with an instance of PA7 as a premise reach $\forall x\mathcal{P}$ by $\to$E. The use of IN merely saves a couple steps, and avoids some relatively long formulas we would have to deal with using P7 alone. Thus, from our previous example, to apply IN we need $\mathcal{P}^x_\emptyset$ and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$ to move to $\forall x\mathcal{P}$. So, if $\mathcal{P}$ is $\sim(x = Sx)$, we would need $\sim(\emptyset = S\emptyset)$ and $\forall x[\sim(x = Sx) \to \sim(Sx = SSx)]$ to move to $\forall x\sim(x = Sx)$ by IN. You should see that this is no different from before.

In this system, there is no need for an axiom like Q7, insofar as we shall be able to derive it with the aid of PA7. That is, for y not in t we shall be able to show,

T6.48. PA $\vdash_{ND} \sim(t = \emptyset) \to \exists y(t = Sy)$

> Since it is to follow from PA1 - PA7, the proof must, of course, not depend on Q7 and so on any of T6.43, T6.46, or T6.47.

But T6.48 has Q7 as an instance. Given this, any *ND* theorem of Q is automatically an *ND* theorem of PA — for we can derive T6.48, and use it as it would have been

used in a derivation for Q. We thus freely use any theorem from Q in the derivations that follow.

With these axioms, including the principle of mathematical induction, in hand we set out to show some general principles of commutativity, associativity and distribution for addition and multiplication. But we build gradually to them. For a first application of IN, let $\mathcal{P}$ be $(\emptyset + x) = x$; then $\mathcal{P}^x_\emptyset$ is $(\emptyset + \emptyset) = \emptyset$ and $\mathcal{P}^x_{Sx}$ is $(\emptyset + Sx) = Sx$.

T6.49. PA $\vdash_{ND}$ $(\emptyset + t) = t$

1.	$(\emptyset + \emptyset) = \emptyset$	T6.39
2.	$(\emptyset + Sj) = S(\emptyset + j)$	T6.40
3.	$\quad (\emptyset + j) = j$	A $(g, \to\text{I})$
4.	$\quad (\emptyset + Sj) = Sj$	2,3 =E
5.	$[(\emptyset + j) = j] \to [(\emptyset + Sj) = Sj]$	3-4 →I
6.	$\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$	5 ∀I
7.	$\forall x[(\emptyset + x) = x]$	1,6 IN
8.	$(\emptyset + t) = t$	7 ∀E

The key to this derivation, and others like it, is bringing IN into play. That we want to do this, is sufficient to drive us to the following as setup.

(CD)

$(\emptyset + \emptyset) = \emptyset$	(goal)
$\quad (\emptyset + j) = j$	A $(g, \to\text{I})$
$\quad (\emptyset + Sj) = Sj$	(goal)
$[(\emptyset + j) = j] \to [(\emptyset + Sj) = Sj]$	→I
$\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$	∀I
$\forall x[(\emptyset + x) = x]$	IN
$(\emptyset + t) = t$	∀E

Our aim is to get the goal by ∀E from $\forall x[(\emptyset + x) = x]$. And we will get this by IN. So we need the inputs to IN, $\mathcal{P}^x_\emptyset$, that is, $(\emptyset + \emptyset) = \emptyset$, and $\forall x(\mathcal{P} \to \mathcal{P}^x_{Sx})$, that is, $\forall x([(\emptyset + x) = x] \to [(\emptyset + Sx) = Sx])$. As is often the case, $\mathcal{P}^x_\emptyset$, here $(\emptyset + \emptyset) = \emptyset$, is easy to get. It is natural to get the latter by ∀I from $[(\emptyset + j) = j] \to [(\emptyset + Sj) = Sj]$, and to go for this by →I. The work of the derivation is reaching our two goals. But that is not hard. The first is an immediate instance of T6.39. And the second follows from the equality on (3), with an instance of T6.40. We are in a better position to *think* about which (axioms or) theorems we need as premises once

we have gone through this standard setup for IN. We will see this pattern over and over.

T6.50. PA $\vdash_{ND} (St + \emptyset) = S(t + \emptyset)$

1.	$(St + \emptyset) = St$	T6.39
2.	$(t + \emptyset) = t$	T6.39
3.	$(St + \emptyset) = S(t + \emptyset)$	1,2 =E

This simple derivation results by using the equality on (2) to justify a substitution for t in (1). This result forms the "zero case" for the one that follows.

T6.51. PA $\vdash_{ND} (St + \mathscr{s}) = S(t + \mathscr{s})$

1.	$(St + \emptyset) = S(t + \emptyset)$	T6.50
2.	$(t + Sj) = S(t + j)$	T6.40
3.	$(St + Sj) = S(St + j)$	T6.40
4.	$\quad (St + j) = S(t + j)$	A (g, $\rightarrow$I)
5.	$\quad (St + Sj) = SS(t + j)$	3,4 =E
6.	$\quad (St + Sj) = S(t + Sj)$	5,2 =E
7.	$[(St + j) = S(t + j)] \rightarrow [(St + Sj) = S(t + Sj)]$	4-6 $\rightarrow$I
8.	$\forall x([(St + x) = S(t + x)] \rightarrow [(St + Sx) = S(t + Sx)])$	7 $\forall$I
9.	$\forall x[(St + x) = S(t + x)]$	1,8 IN
10.	$(St + \mathscr{s}) = S(t + \mathscr{s})$	9 $\forall$E

Again, the idea is to bring IN into play. Here $\mathscr{P}$ is $(St+x) = S(t+x)$. Given that we have the zero-case on line (1), with standard setup, the derivation reduces to obtaining the formula on (6) given the assumption on (4). Line (6) is like (3) except for the right-hand side. So it is a matter of applying the equalities on (4) and (2) to reach the goal. You should study this derivation, to be sure that you follow the applications of =E. If you do, you are managing some reasonably complex applications of the rule!

T6.52. PA $\vdash_{ND}$ $(t + s) = (s + t)$ *commutativity of addition*

1.	$(t + \emptyset) = t$	T6.39
2.	$(\emptyset + t) = t$	T6.49
3.	$(t + Sj) = S(t + j)$	T6.40
4.	$(Sj + t) = S(j + t)$	T6.51
5.	$(t + \emptyset) = (\emptyset + t)$	1,2 =E
6.	$\quad (t + j) = (j + t)$	A (g, →I)
7.	$\quad (t + Sj) = S(j + t)$	3,6 =E
8.	$\quad (t + Sj) = (Sj + t)$	7,4 =E
9.	$[(t + j) = (j + t)] \to [(t + Sj) = (Sj + t)]$	6-8 →I
10.	$\forall x([(t + x) = (x + t)] \to [(t + Sx) = (Sx + t)])$	9 ∀I
11.	$\forall x[(t + x) = (x + t)]$	5,10 IN
12.	$(t + s) = (s + t)$	11 ∀E

Again the derivation is by IN where $\mathscr{P}$ is $(t+x) = (x+t)$. We achieve the zero case on (5) from (1) and (2). So the derivation reduces to getting (8) given the assumption on (6). The left-hand side of (8) is like (3). So it is a matter of applying the equalities on (6) and then (4) to reach the goal. Very often the challenge in these cases is not so much doing the derivations, as organizing in your mind which equalities you have, and which are required to reach the goal.

T6.52 is an interesting result! No doubt, you have heard from your mother's knee that $(t + s) = (s + t)$. But it is a sweeping claim with application to *all* numbers. Surely you have not been able to test every case. But here we have a derivation of the result, from the Peano Axioms. And similarly for results that follow. Now that you have this result, recognize that you can use instances of it to switch around terms in additions — just as you would have done automatically for addition in elementary school.

*T6.53. PA $\vdash_{ND}$ $[(r + s) + \emptyset] = [r + (s + \emptyset)]$

Hint: Begin with $((r + s) + \emptyset) = (r + s)$ as an instance of T6.39. The derivation is then a simple matter of using T6.39 again to "replace" s in the right-hand side with $s + \emptyset$.

*T6.54. PA $\vdash_{ND}$ $[(r + s) + t] = [r + (s + t)]$ *associativity of addition*

Hint: For an application of IN let $\mathscr{P}$ be $[(r + s) + x] = [r + (s + x)]$. You already have the zero case from T6.53. Inside the subderivation for →I, use the assumption together with some instances of T6.40 to reach the goal.

Again, once you have this result, be aware that you can use its instances for association as you would have done long ago. It is good to *think* about what the different theorems give you, so that you can make sense of what to use where!

T6.55. PA $\vdash_{ND}$ $(t \times S\emptyset) = t$

Hint: This does not require IN. It is a rather a simple result which you can do in just five lines.

T6.56. PA $\vdash_{ND}$ $(\emptyset \times t) = \emptyset$

Hint: For an application of IN, let $\mathscr{P}$ be $(\emptyset \times x) = \emptyset$. The derivation is easy enough with an application of T6.41 for the zero case, and instances of T6.42 and T6.39 for the main result.

T6.57. PA $\vdash_{ND}$ $(St \times \emptyset) = [(t \times \emptyset) + \emptyset]$

Hint: This does not require IN. It follows rather by some simple applications of T6.39 and T6.41.

T6.58. PA $\vdash_{ND}$ $(St \times s) = [(t \times s) + s]$

Hint: For this longish derivation, plan to reach the goal through IN where $\mathscr{P}$ is $(St \times x) = [(t \times x) + x]$. You will be able to use your assumption for $\rightarrow$I with an instance of T6.42 to show $(St \times Sj) = ([(t \times j) + j] + St)$. And you should be able to use associativity and the like to manipulate the right-hand side into the result you want. You will need several theorems as premises.

T6.59. PA $\vdash_{ND}$ $(t \times s) = (s \times t)$ *commutativity for multiplication*

Hint: Plan on reaching the goal by IN where $\mathscr{P}$ is $(t \times x) = (x \times t)$. Apart from theorems for the zero case, you will need an instance of T6.42, and an instance of T6.58.

T6.60. PA $\vdash_{ND}$ $[r \times (s + \emptyset)] = [(r \times s) + (r \times \emptyset)]$

Hint: You will not need IN for this.

T6.61. PA $\vdash_{ND}$ $[r \times (s + t)] = [(r \times s) + (r \times t)]$ *distributivity*

Hint: Plan on reaching the goal by IN where $\mathscr{P}$ is $[r \times (s + x)] = [(r \times s) + (r \times x)]$. Perhaps the simplest thing is to start with $[r \times (s + Sj)] = [r \times (s + Sj)]$ by =I. Then the left side is what you want, and you can work on the right. Working on the right-hand side, $(s + Sj) = S(s + j)$ by T6.40. And $[r \times S(s + j)] = ([r \times (s + j)] + r)$ by T6.42. With this, you will be able to apply the assumption for →I. And further simplification should get you to your goal.

T6.62. PA $\vdash_{ND}$ $[(s + t) \times r] = [(s \times r) + (t \times r)]$ *distributivity*

Hint: You will not need IN for this. Rather, it is enough to use T6.61 with a few applications of T6.59.

T6.63. PA $\vdash_{ND}$ $(r + s) \times (t + u) = [(r \times s) + (r \times u)] + [(s \times t) + s \times u)]$

Hint: This is a simple application of distributivity.

T6.64. PA $\vdash_{ND}$ $[(s \times t) \times \emptyset] = [s \times (t \times \emptyset)]$

Hint: This is easy without an application of IN.

T6.65. PA $\vdash_{ND}$ $[(s \times t) \times r] = [s \times (t \times r)]$ *associativity of multiplication*

Hint: Go after the goal by IN where $\mathscr{P}$ is $[(s \times t) \times x] = [s \times (t \times x)]$. You should be able to use the assumption with T6.42 to show that $[(s \times t) \times Sj] = [(s \times (t \times j)) + (s \times t)]$; then you can reduce the right hand side to what you want.

T6.66. PA $\vdash_{ND}$ $(r + t = s + t) \to r = s$ *cancellation law for addition*

T6.67. PA $\vdash_{ND}$ $(s \neq \emptyset \land t \times s = r \times s) \to t = r$ *cancellation law for multiplication*

Robinson and Peano Arithmetic (ND)

Q/PA 1. $\sim(Sx = \emptyset)$

2. $(Sx = Sy) \rightarrow (x = y)$

3. $(x + \emptyset) = x$

4. $(x + Sy) = S(x + y)$

5. $(x \times \emptyset) = \emptyset$

6. $(x \times Sy) = [(x \times y) + x]$

Q7 $\sim(x = \emptyset) \rightarrow \exists y(x = Sy)$

PA7 $[\mathcal{P}^x_\emptyset \wedge \forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})] \rightarrow \forall x\mathcal{P}$

IN

a. $\mathcal{P}^x_\emptyset$

b. $\forall x(\mathcal{P} \rightarrow \mathcal{P}^x_{Sx})$

$\forall x\mathcal{P}$ a,b IN

Derived from PA7

T6.37 Q $\vdash_{ND}$ $\sim(St = \emptyset)$

T6.38 Q $\vdash_{ND}$ $(St = Ss) \rightarrow (t = s)$

T6.39 Q $\vdash_{ND}$ $(t + \emptyset) = t$

T6.40 Q $\vdash_{ND}$ $(t + Ss) = S(t + s)$

T6.41 Q $\vdash_{ND}$ $(t \times \emptyset) = \emptyset$

T6.42 Q $\vdash_{ND}$ $(t \times Ss) = ((t \times s) + t)$

T6.43 Q $\vdash_{ND}$ $\sim(t = \emptyset) \rightarrow \exists y(t = Sy)$ where variable y does not appear in t

T6.44 Q $\vdash_{ND}$ $S\emptyset + S\emptyset = SS\emptyset$

T6.45 Q $\vdash_{ND}$ $t + S\emptyset = St$

T6.46 Q $\vdash_{ND}$ $\forall x(x \leq \emptyset \rightarrow x = \emptyset)$

T6.47 Q $\vdash_{ND}$ $\forall x\sim(x < \emptyset)$

T6.48 PA $\vdash_{ND}$ $\sim(t = \emptyset) \rightarrow \exists y(t = Sy)$ *(y not in t) and so Q7*

T6.49 PA $\vdash_{ND}$ $(\emptyset + t) = t$

T6.50 PA $\vdash_{ND}$ $(St + \emptyset) = S(t + \emptyset)$

T6.51 PA $\vdash_{ND}$ $(St + s) = S(t + s)$

T6.52 PA $\vdash_{ND}$ $(t + s) = (s + t)$ *commutativity of addition*

T6.53 PA $\vdash_{ND}$ $[(r + s) + \emptyset] = [r + (s + \emptyset)]$

T6.54 PA $\vdash_{ND}$ $[(r + s) + t] = [r + (s + t)]$ *associativity of addition*

T6.55 PA $\vdash_{ND}$ $(t \times S\emptyset) = t$

T6.56 PA $\vdash_{ND}$ $(\emptyset \times t) = \emptyset$

T6.57 PA $\vdash_{ND}$ $(St \times \emptyset) = [(t \times \emptyset) + \emptyset]$

T6.58 PA $\vdash_{ND}$ $(St \times s) = [(t \times s) + s]$

T6.59 PA $\vdash_{ND}$ $(t \times s) = (s \times t)$ *commutativity for multiplication*

T6.60 PA $\vdash_{ND}$ $[r \times (s + \emptyset)] = [(r \times s) + (r \times \emptyset)]$

T6.61 PA $\vdash_{ND}$ $[r \times (s + t)] = [(r \times s) + (r \times t)]$ *distributivity*

T6.62 PA $\vdash_{ND}$ $[(s + t) \times r] = [(s \times r) + (t \times r)]$ *distributivity*

T6.63 PA $\vdash_{ND}$ $(r + s) \times (t + u) = [(r \times s) + (r \times u)] + [(s \times t) + s \times u)]$

T6.64 PA $\vdash_{ND}$ $[(s \times t) \times \emptyset] = [s \times (t \times \emptyset)]$

T6.65 PA $\vdash_{ND}$ $[(s \times t) \times r] = [s \times (t \times r)]$ *associativity of multiplication*

T6.66 PA $\vdash_{ND}$ $(r + t = s + t) \rightarrow r = s$ *cancellation law for addition*

T6.67 PA $\vdash_{ND}$ $(s \neq \emptyset \wedge t \times s = r \times s) \rightarrow t = r$ *cancellation law for multiplication*

After you have completed the exercises, if you are looking for more to do, you might take a look at the additional results from T13.13 on p. 622 — or, really, once you get started all of section 13.2 - 13.6 is a playground for proofs in PA.

Peano Arithmetic is thus sufficient for results we could not obtain in Q alone — for "ordinary" arithmetic. However, insofar as it includes the language and results of Q, it too is sufficient for Gödel's incompleteness theorem. So PA is not complete, and it is not possible for a nicely specified theory including PA to be such that it proves either $\mathcal{P}$ or $\sim\mathcal{P}$ for every $\mathcal{P}$. But such results must wait for later.

*E6.33. Produce derivations to show T6.31 - T6.36. Hint: it may help to begin with concrete versions of the theorems and then move to the general case. Thus, for example, for T6.32, show that $\vdash_{ND} (y = j) \rightarrow (g^3xyz = g^3xjz)$. Then you will be able to show the general case.

E6.34. Produce derivations to show each of the following.

*a. $\vdash_{ND} \forall x\exists y(x = y)$

b. $\vdash_{ND} \forall x\exists y(f^1x = y)$

c. $\vdash_{ND} \forall x\forall y[(Fx \wedge \sim Fy) \rightarrow \sim(x = y)]$

d. $\forall x(Rxa \rightarrow x = c), \forall x(Rxb \rightarrow x = d), \exists x(Rxa \wedge Rxb) \vdash_{ND} c = d$

e. $\vdash_{ND} \forall x[\sim(f^1x = x) \rightarrow \forall y((f^1x = y) \rightarrow \sim(x = y))]$

f. $\vdash_{ND} \forall x\forall y[(f^1x = y \wedge f^1y = x) \rightarrow f^1f^1x = x]$

g. $\exists x\exists yHxy, \forall y\forall z(Dyz \leftrightarrow Hzy), \forall x\forall y(\sim Hxy \vee x = y)$
$\vdash_{ND} \exists x(Hxx \wedge Dxx)$

h. $\forall x\forall y[(Rxy \wedge Ryx) \rightarrow x = y], \forall x\forall y(Rxy \rightarrow Ryx)$
$\vdash_{ND} \forall x[\exists y(Rxy \vee Ryx) \rightarrow Rxx]$

i. $\exists x\forall y(x = y \leftrightarrow Fy), \forall x(Gx \rightarrow Fx) \vdash_{ND} \forall x\forall y[(Gx \wedge Gy) \rightarrow x = y]$

j. $\forall x[Fx \rightarrow \exists y(Gyx \wedge \sim Gxy)], \forall x\forall y[(Fx \wedge Fy) \rightarrow x = y]$
$\vdash_{ND} \forall x(Fx \rightarrow \exists y\sim Fy)$

k. $\exists xFx, \forall x\forall y[x = y \vee \sim(Fx \wedge Fy)] \vdash_{ND} \exists x\forall y(x = y \leftrightarrow Fy)$

*E6.35. Produce derivations to show derived rules for the bounded quantifiers along with T6.39 - T6.43, T6.45 - T6.47 and each of the following. You should hold off on derivations for T6.46 and T6.47 until the end. For any problem, you may appeal to results before.

*a. Q $\vdash_{ND}$ $(SS\emptyset + S\emptyset) = SSS\emptyset$

b. Q $\vdash_{ND}$ $(SS\emptyset + SS\emptyset) = SSSS\emptyset$

c. Q $\vdash_{ND}$ $(\emptyset + S\emptyset) = S\emptyset$

d. Q $\vdash_{ND}$ $(S\emptyset \times S\emptyset) = S\emptyset$

e. Q $\vdash_{ND}$ $(SS\emptyset \times SS\emptyset) = SSSS\emptyset$

Hint: You may decide some preliminary results will be helpful.

*f. Q $\vdash_{ND}$ $\sim\exists x(x + SS\emptyset = S\emptyset)$

Hint: Do not forget that you can appeal to T6.37 and T6.38.

g. Q $\vdash_{ND}$ $\forall x[(x = \emptyset \vee x = S\emptyset) \rightarrow x \leq S\emptyset)]$

h. Q $\vdash_{ND}$ $\forall x[(x = \emptyset \vee x = S\emptyset) \rightarrow x < SS\emptyset)]$

i. Q $\vdash_{ND}$ $(\forall x \leq S\emptyset)(x = \emptyset \vee x = S\emptyset)$

Hint: You will be able to use T6.46 to show that if $a + b = \emptyset$ then $b = \emptyset$.

j. Q $\vdash_{ND}$ $(\forall x \leq S\emptyset)(x \leq SS\emptyset)$

Hint: You may find the previous result helpful.

*E6.36. Produce derivations to show T6.53 - T6.67.

E6.37. Produce a derivation to show T6.48 and so that any ND theorem of Q is an ND theorem of PA. Hint: For an application of IN let $\mathscr{P}$ be $\sim(x = \emptyset) \rightarrow \exists y(x = Sy)$.

6.4 The system *ND+*

ND+ includes all the rules of *ND*, with four new inference rules, and some new *replacement* rules. It is not possible to derive anything in *ND+* that cannot already be derived in *ND*. Thus the new rules do not add extra derivation power. They are rather "shortcuts" for things that can already be done in *ND*. This is particularly obvious in the case of the inference rules.

For the first, suppose in an *ND* derivation, we have $\mathcal{P} \to \mathcal{Q}$ and $\sim\mathcal{Q}$ and want to reach $\sim\mathcal{P}$. No doubt, we would proceed as follows.

(CE)

1.	$\mathcal{P} \to \mathcal{Q}$	P
2.	$\sim\mathcal{Q}$	P
3.	$\quad \mathcal{P}$	A (c, $\sim$I)
4.	$\quad \mathcal{Q}$	1,3 $\to$E
5.	$\quad \bot$	4,1 $\bot$I
6.	$\sim\mathcal{P}$	3-5 $\sim$I

We assume $\mathcal{P}$, get the contradiction, and conclude by $\sim$I. Perhaps you have done this so many times that you can do it in your sleep. In *ND+* you are given a way to shortcut the routine, and go directly from an accessible $\mathcal{P} \to \mathcal{Q}$ on a, and an accessible $\sim\mathcal{Q}$ on b to $\sim\mathcal{P}$ with justification *a,b* MT (*modus tollens*).

MT

a	$\mathcal{P} \to \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\sim\mathcal{P}$	a,b MT

The justification for this is that the rule does not let you do anything that you could not already do in *ND*. So if the rules of *ND* preserve truth, this rule preserves truth. And, as a matter of fact, we already demonstrated that $\mathcal{P} \to \mathcal{Q}, \sim\mathcal{Q} \vdash_{ND} \sim\mathcal{P}$ in T6.4. Similarly, T6.5, T6.6, T6.7, T6.8 and T6.9 justify the other inference rules included in *ND+*.

NB

a	$\mathcal{P} \leftrightarrow \mathcal{Q}$		a	$\mathcal{P} \leftrightarrow \mathcal{Q}$	
b	$\sim\mathcal{P}$		b	$\sim\mathcal{Q}$	
	$\sim\mathcal{Q}$	a,b NB		$\sim\mathcal{P}$	a,b NB

NB (*negated biconditional*) lets you move from a biconditional and the negation of one side, to the negation of the other. It is like MT, but with the arrow going both ways. The parts are justified in T6.8 and T6.9.

DS

a	$\mathcal{P} \vee \mathcal{Q}$	
b	$\sim\mathcal{P}$	
	$\mathcal{Q}$	a,b DS

a	$\mathcal{P} \vee \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\mathcal{P}$	a,b DS

DS (*disjunctive syllogism*) lets you move from a disjunction and the negation of one side, to the other side of the disjunction. We saw an intuitive version of this rule on p. 26. The two parts are justified by T6.6 and T6.7.

HS

a	$\mathcal{O} \rightarrow \mathcal{P}$	
b	$\mathcal{P} \rightarrow \mathcal{Q}$	
	$\mathcal{O} \rightarrow \mathcal{Q}$	a,b HS

HS (*hypothetical syllogism*) is a principle of transitivity by which you may string a pair of conditionals together into one. It is justified by T6.5.

Each of these rules should be clear, and easy to use. Here is an example that puts all of them together into one derivation.

(CF)

1.	$A \leftrightarrow B$	P
2.	$\sim B$	P
3.	$A \vee (C \rightarrow D)$	P
4.	$D \rightarrow B$	P
5.	$\sim A$	1,2 NB
6.	$C \rightarrow D$	3,5 DS
7.	$C \rightarrow B$	6,4 HS
8.	$\sim C$	7,2 MT

1.	$A \leftrightarrow B$	P
2.	$\sim B$	P
3.	$A \vee (C \rightarrow D)$	P
4.	$D \rightarrow B$	P
5.	A	A (g, (3$\vee$E)
7.	C	A (c, $\sim$I)
8.	B	1,5 $\leftrightarrow$E
9.	$\perp$	8,2 $\perp$I
10.	$\sim C$	6-8 $\sim$I
10.	$C \rightarrow D$	A (g, 3$\vee$E)
11.	C	A (c, $\sim$I)
12.	D	10,11 $\sim$E
13.	B	4,12 $\rightarrow$E
14.	$\perp$	13,2 $\perp$I
15.	$\sim C$	11-14 $\sim$I
16.	$\sim C$	3,5-9,10-15 $\vee$E

We can do it by our normal methods with the rules of *ND* as on the right. But it is easier with the shortcuts from *ND+* as on the left. It may take you some time to “see” applications of the new rules when you are doing derivations, but the simplification makes it worth getting used to them.

The replacement rules of *ND+* are different from ones we have seen before in two respects. First, replacement rules go in two directions. Consider the following simple rule.

DN $\qquad \mathcal{P} \quad \triangleleft\triangleright \quad \sim\sim\mathcal{P}$

According to DN (*double negation*), given $\mathcal{P}$ on an accessible line a, you may move to $\sim\sim\mathcal{P}$ with justification a DN; and given $\sim\sim\mathcal{P}$ on an accessible line a, you may move to $\mathcal{P}$ with justification a DN. This two-way rule is justified by T6.16, in which we showed $\vdash_{ND} \mathcal{P} \leftrightarrow \sim\sim\mathcal{P}$. Given $\mathcal{P}$ we could use the routine from one half of the derivation to reach $\sim\sim\mathcal{P}$, and given $\sim\sim\mathcal{P}$ we could use the routine from the other half of the derivation to reach $\mathcal{P}$.

But, further, we can use replacement rules to replace a subformula that is just a proper part of another formula. Thus, for example, in the following list, we could move in one step by DN from the formula on the left, to any of the ones on the right, and from any of the ones on the right, to the one on the left.

		$\sim\sim[A \wedge (B \to C)]$
		$\sim\sim A \wedge (B \to C)$
(CG)	$A \wedge (B \to C)$	$A \wedge \sim\sim(B \to C)$
		$A \wedge (\sim\sim B \to C)$
		$A \wedge (B \to \sim\sim C)$

The first application is of the sort we have seen before, in which the whole formula is replaced. In the second, the replacement is between the subformulas A and $\sim\sim A$. In the third, between the subformulas $(B \to C)$ and $\sim\sim(B \to C)$. The fourth switches B and $\sim\sim B$ and the last C and $\sim\sim C$. Thus the DN rule allows the substitution of any subformula $\mathcal{P}$ with one of the form $\sim\sim\mathcal{P}$, and vice versa.

The application of replacement rules to subformulas is not so easily justified as their application to whole formulas. A complete justification that *ND+* does not let you go beyond what can be derived in *ND* will have to wait for Part III. Roughly, though, the idea is this: given a complex formula, we can take it apart, do the replacement, and then put it back together. Here is a very simple example from above.

(CH)

1.	$A \wedge (B \to C)$	P
2.	$A \wedge \sim\sim(B \to C)$	1 DN

1.	$A \wedge (B \to C)$	P
2.	A	1 $\wedge$E
3.	$\quad \sim(B \to C)$	A (c, $\sim$I)
4.	$\quad B \to C$	1 $\wedge$E
5.	$\quad \bot$	4,3 $\bot$I
6.	$\sim\sim(B \to C)$	3-5 $\sim$I
7.	$A \wedge \sim\sim(B \to C)$	2,6 $\wedge$I

On the left, we make the move from $A \wedge (B \rightarrow C)$ to $A \wedge \sim\sim(B \rightarrow C)$ in one step by DN. On the right, using just the rules of *ND*, we begin by taking off the A. Then we convert $B \rightarrow C$ to $\sim\sim(B \rightarrow C)$, and put it back together with the A. Though we will not be able to show that sort of thing is generally possible until Part III, for now I will continue to say that replacement rules are "justified" by the corresponding biconditionals. As it happens, for replacement rules, the biconditionals play a crucial role in the demonstration that $\Gamma \vdash_{ND} \mathcal{P}$ iff $\Gamma \vdash_{ND+} \mathcal{P}$.

The rest of the replacement rules work the same way.

Com $\quad \mathcal{P} \wedge \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{Q} \wedge \mathcal{P}$
$\quad \mathcal{P} \vee \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{Q} \vee \mathcal{P}$

Com (*commutation*) lets you reverse the order of conjuncts or disjuncts around an operator. By Com you could go from, say, $A \wedge (B \vee C)$ to $(B \vee C) \wedge A$, switching the order around $\wedge$, or from $A \wedge (B \vee C)$ to $A \wedge (C \vee B)$, switching the order around $\vee$. You should be clear about why this is so. The two forms are justified by T6.10 and T6.11.

Assoc $\quad \mathcal{O} \wedge (\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \wedge \mathcal{Q}$
$\quad \mathcal{O} \vee (\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \vee \mathcal{P}) \vee \mathcal{Q}$

Assoc (*association*) lets you shift parentheses for conjoined or disjoined formulas. The two forms are justified by T6.14 and T6.15.

Idem $\quad \mathcal{P} \;\triangleleft\triangleright\; \mathcal{P} \wedge \mathcal{P}$
$\quad \mathcal{P} \;\triangleleft\triangleright\; \mathcal{P} \vee \mathcal{P}$

Idem (*idempotence)* exposes the equivalence between $\mathcal{P}$ and $\mathcal{P} \wedge \mathcal{P}$, and between $\mathcal{P}$ and $\mathcal{P} \vee \mathcal{P}$. The two forms are justified by T6.17 and T6.18.

Impl $\quad \mathcal{P} \rightarrow \mathcal{Q} \;\triangleleft\triangleright\; \sim\mathcal{P} \vee \mathcal{Q}$
$\quad \sim\mathcal{P} \rightarrow \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{P} \vee \mathcal{Q}$

Impl (*implication*) lets you move between a conditional and a corresponding disjunction. Thus, for example, by the first form of Impl you could move from $A \rightarrow (\sim B \vee C)$ to $\sim A \vee (\sim B \vee C)$, using the rule from left-to-right, or to $A \rightarrow (B \rightarrow C)$, using the rule from right-to-left. As we will see, this rule can be particularly useful. The two forms are justified by T6.21 and T6.22.

Trans $\quad \mathcal{P} \rightarrow \mathcal{Q} \;\triangleleft\triangleright\; \sim\mathcal{Q} \rightarrow \sim\mathcal{P}$

Trans (*transposition*) lets you reverse the antecedent and consequent around a conditional — subject to the addition or removal of negations. From left-to-right, this rule should remind you of MT, as Trans plus $\rightarrow$E has the same effect as one application of MT. Trans is justified by T6.12.

DeM $\sim(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \sim\mathcal{P} \vee \sim\mathcal{Q}$
$\sim(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \sim\mathcal{P} \wedge \sim\mathcal{Q}$

DeM (*DeMorgan*) should remind you of equivalences we learned in chapter 5, for *not both* (the first form) and *neither nor* (the second form). This rule also can be very useful. The two forms are justified by T6.19 and T6.20.

Exp $\mathcal{O} \to (\mathcal{P} \to \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \to \mathcal{Q}$

Exp (*exportation*) is another equivalence that may have arisen in translation. It is justified by T6.13.

Equiv $\mathcal{P} \leftrightarrow \mathcal{Q} \;\triangleleft\triangleright\; (\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P})$
$\mathcal{P} \leftrightarrow \mathcal{Q} \;\triangleleft\triangleright\; (\mathcal{P} \wedge \mathcal{Q}) \vee (\sim\mathcal{P} \wedge \sim\mathcal{Q})$

Equiv (*equivalence*) converts between a biconditional, and the corresponding pair of conditionals, or converts between a biconditional and a formula on which the sides are both true or both false. The two forms are justified by T6.25 and T6.26.

Dist $\mathcal{O} \wedge (\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \vee (\mathcal{O} \wedge \mathcal{Q})$
$\mathcal{O} \vee (\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \vee \mathcal{P}) \wedge (\mathcal{O} \vee \mathcal{Q})$

Dist (*distribution*) works something like the mathematical principle for multiplying across a sum. In each case, moving from left to right, the operator from outside attaches to each of the parts inside the parenthesis, and the operator from inside becomes the main operator. The two forms are justified by T6.23 and T6.24.

QN $\sim\forall x\mathcal{P} \;\triangleleft\triangleright\; \exists x\sim\mathcal{P}$
$\sim\exists x\mathcal{P} \;\triangleleft\triangleright\; \forall x\sim\mathcal{P}$

QN (*quantifier negation*) is another principle we encountered in chapter 5. It lets you push or pull a negation across a quantifier, with a corresponding flip from one quantifier to the other. The forms are justified by T6.29 and T6.30.

BQN $\sim(\forall x \le t)\mathcal{P} \;\triangleleft\triangleright\; (\exists x \le t)\sim\mathcal{P}$ $\sim(\forall x < t)\mathcal{P} \;\triangleleft\triangleright\; (\exists x < t)\sim\mathcal{P}$
$\sim(\exists x \le t)\mathcal{P} \;\triangleleft\triangleright\; (\forall x \le t)\sim\mathcal{P}$ $\sim(\exists x < t)\mathcal{P} \;\triangleleft\triangleright\; (\forall x < t)\sim\mathcal{P}$

BQN (*bounded quantifier negation*) applies to the abbreviations introduced on p. 301. It works by analogy with QN. Its demonstration requires a new theorem:

T6.68. The following result in PA:

(a) $\vdash_{ND} \sim(\forall x \le t)\mathcal{P} \leftrightarrow (\exists x \le t)\sim\mathcal{P}$

(b) $\vdash_{ND} \sim(\exists x \le t)\mathcal{P} \leftrightarrow (\forall x \le t)\sim\mathcal{P}$.

(c) $\vdash_{ND} \sim(\forall x < t)\mathcal{P} \leftrightarrow (\exists x < t)\sim\mathcal{P}$

(d) $\vdash_{ND} \sim(\exists x < t)\mathcal{P} \leftrightarrow (\forall x < t)\sim\mathcal{P}$.

Demonstration of this result is left to E6.39.

Thus end the rules of *ND+*. They are a lot to absorb at once. But you do not need to absorb all the rules at once. Again, the rules do not let you do anything you could not already do in *ND*. For the most part, you should proceed as if you were in *ND*. If an *ND+* shortcut occurs to you, use it. You will gradually become familiar with more and more of the special *ND+* rules. Perhaps, though, we can make a few observations about strategy that will get you started. First, again, do not get too distracted by the extra rules! You should continue with the overall goal-directed approach from *ND*. There are, however, a few contexts where special rules from *ND+* can make a substantive difference. I comment on three.

First, as we have seen, in *ND*, formulas with $\vee$ can be problematic. $\vee$E is awkward to apply, and $\vee$I does not always work. In simple cases, DS can get you out of $\vee$E. But this is not always so, and you will want to keep $\vee$E among your standard strategies. More importantly, Impl can convert between awkward formulas with main operator $\vee$ and more manageable ones with main operator $\rightarrow$. For premises, this does not help much. DS gets you just as much as Impl and then $\rightarrow$E or MT (think about it). But converting to $\rightarrow$ does matter when a *goal* has main operator $\vee$. Although a disjunction may be derivable, but not by $\vee$I, if a conditional is derivable, it *is* derivable by $\rightarrow$I. Thus to reach a goal with main operator $\vee$, consider going for the corresponding $\rightarrow$, and converting with Impl.

given

	$\mathcal{A} \vee \mathcal{B}$	(goal)

use

a.	$\sim\mathcal{A}$	A (g $\rightarrow$I)
b.	$\mathcal{B}$	(goal)
	$\sim\mathcal{A} \rightarrow \mathcal{B}$	$\rightarrow$I
	$\mathcal{A} \vee \mathcal{B}$	Impl

And the other form of Impl may be helpful for a goal of the sort $\sim\mathcal{A} \vee \mathcal{B}$. Here is a quick example.

(CI)

1.	$\sim A$	A (g, $\rightarrow$I)
2.	$\sim A$	1 R
3.	$\sim A \rightarrow \sim A$	1-2 $\rightarrow$I
4.	$A \vee \sim A$	3 Impl

1.	$\sim(A \vee \sim A)$	A (c, $\sim$E)
2.	A	A (c, $\sim$I)
3.	$A \vee \sim A$	2 $\vee$I
4.	$\bot$	4,1 $\bot$I
5.	$\sim A$	2-4 $\sim$I
6.	$A \vee \sim A$	5 $\vee$I
7.	$\bot$	6,1 $\bot$I
8.	$A \vee \sim A$	1-7 $\sim$E

ND+ Quick Reference

Inference Rules

MT (*Modus Tollens*)

a	$\mathcal{P} \rightarrow \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\sim\mathcal{P}$	a,b MT

NB (*Negated Biconditional*)

a	$\mathcal{P} \leftrightarrow \mathcal{Q}$	
b	$\sim\mathcal{P}$	
	$\sim\mathcal{Q}$	a,b NB

NB (*Negated Biconditional*)

a	$\mathcal{P} \leftrightarrow \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\sim\mathcal{P}$	a,b NB

DS (*Disjunctive Syllogism*)

a	$\mathcal{P} \vee \mathcal{Q}$	
b	$\sim\mathcal{P}$	
	$\mathcal{Q}$	a,b DS

DS (*Disjunctive Syllogism*)

a	$\mathcal{P} \vee \mathcal{Q}$	
b	$\sim\mathcal{Q}$	
	$\mathcal{P}$	a,b DS

HS (*Hypothetical Syllogism*)

a	$\mathcal{O} \rightarrow \mathcal{P}$	
b	$\mathcal{P} \rightarrow \mathcal{Q}$	
	$\mathcal{O} \rightarrow \mathcal{Q}$	a,b HS

Replacement Rules

DN	$\mathcal{P} \;\triangleleft\triangleright\; \sim\sim\mathcal{P}$
Com	$\mathcal{P} \wedge \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{Q} \wedge \mathcal{P}$ $\mathcal{P} \vee \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{Q} \vee \mathcal{P}$
Assoc	$\mathcal{O} \wedge (\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \wedge \mathcal{Q}$ $\mathcal{O} \vee (\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \vee \mathcal{P}) \vee \mathcal{Q}$
Idem	$\mathcal{P} \;\triangleleft\triangleright\; \mathcal{P} \wedge \mathcal{P}$ $\mathcal{P} \;\triangleleft\triangleright\; \mathcal{P} \vee \mathcal{P}$
Impl	$\mathcal{P} \rightarrow \mathcal{Q} \;\triangleleft\triangleright\; \sim\mathcal{P} \vee \mathcal{Q}$ $\sim\mathcal{P} \rightarrow \mathcal{Q} \;\triangleleft\triangleright\; \mathcal{P} \vee \mathcal{Q}$
Trans	$\mathcal{P} \rightarrow \mathcal{Q} \;\triangleleft\triangleright\; \sim\mathcal{Q} \rightarrow \sim\mathcal{P}$
DeM	$\sim(\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; \sim\mathcal{P} \vee \sim\mathcal{Q}$ $\sim(\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; \sim\mathcal{P} \wedge \sim\mathcal{Q}$
Exp	$\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \rightarrow \mathcal{Q}$
Equiv	$\mathcal{P} \leftrightarrow \mathcal{Q} \;\triangleleft\triangleright\; (\mathcal{P} \rightarrow \mathcal{Q}) \wedge (\mathcal{Q} \rightarrow \mathcal{P})$ $\mathcal{P} \leftrightarrow \mathcal{Q} \;\triangleleft\triangleright\; (\mathcal{P} \wedge \mathcal{Q}) \vee (\sim\mathcal{P} \wedge \sim\mathcal{Q})$
Dist	$\mathcal{O} \wedge (\mathcal{P} \vee \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \wedge \mathcal{P}) \vee (\mathcal{O} \wedge \mathcal{Q})$ $\mathcal{O} \vee (\mathcal{P} \wedge \mathcal{Q}) \;\triangleleft\triangleright\; (\mathcal{O} \vee \mathcal{P}) \wedge (\mathcal{O} \vee \mathcal{Q})$
QN	$\sim\forall x\mathcal{P} \;\triangleleft\triangleright\; \exists x\sim\mathcal{P}$ $\sim\exists x\mathcal{P} \;\triangleleft\triangleright\; \forall x\sim\mathcal{P}$
BQN	$\sim(\forall x \leq t)\mathcal{P} \;\triangleleft\triangleright\; (\exists x \leq t)\sim\mathcal{P}$ $\sim(\exists x \leq t)\mathcal{P} \;\triangleleft\triangleright\; (\forall x \leq t)\sim\mathcal{P}$
	$\sim(\forall x < t)\mathcal{P} \;\triangleleft\triangleright\; (\exists x < t)\sim\mathcal{P}$ $\sim(\exists x < t)\mathcal{P} \;\triangleleft\triangleright\; (\forall x < t)\sim\mathcal{P}$

The derivation on the left using Impl is completely trivial, requiring just a derivation of $\sim A \to \sim A$. But the derivation on the right is not. It falls through to SG5, and then requires a challenging application of SC3 or SC4. This proposed strategy replaces or simplifies the pattern (AQ) for disjunctions described on p. 265. Observe that the *work* — getting to one side of a disjunction from the negation of the other, is exactly the same. It is only that we use the derived rule to simplify away the distracting and messy setup.

Second, among the most useless formulas for exploitation are ones with main operator $\sim$. But the combination of QN, DeM, Impl, and Equiv let you "push" negations into arbitrary formulas. Thus you can convert formulas with main operator $\sim$ into a more useful form. To see how these rules can be manipulated, consider the following sequence.

(CJ)

1.	$\sim\exists x(Ax \to Bx)$	P
2.	$\forall x\sim(Ax \to Bx)$	1 QN
3.	$\forall x\sim(\sim Ax \vee Bx)$	2 Impl
4.	$\forall x(\sim\sim Ax \wedge \sim Bx)$	3 DeM
5.	$\forall x(Ax \wedge \sim Bx)$	4 DN

We begin with the negation as main operator, and end with a negation only against an atomic. This sort of thing is often very useful. For example, in going for a contradiction, you have the option of "breaking down" a formula with main operator $\sim$ rather than automatically building up to its opposite, according to SC3. And other strategies can be affected as well. Thus, for example, if you see a negated universal on some accessible line, you should think of it as if it were an existentially quantified expression: push the negation through, get the existential, and go for the goal by $\exists$E as usual. Here is an example.

(ZZ)

1.	$\sim\forall x(Fx \to Gx)$	P
2.	$\exists x\sim(Fx \to Gx)$	1 QN
3.	$\quad\sim(Fj \to Gj)$	A (g, 2$\exists$E)
4.	$\quad\sim(\sim Fj \vee Gj)$	3 Impl
5.	$\quad\sim\sim Fj \wedge \sim Gj$	4 DeM
6.	$\quad\sim Gj$	5 $\wedge$E
7.	$\quad\exists x\sim Gx$	6 $\exists$I
8.	$\exists x\sim Gx$	2,3-7 $\exists$E

1.	$\sim\forall x(Fx \to Gx)$	P
2.	$\quad\sim\exists x\sim Gx$	A (c $\sim$E)
3.	$\quad\quad Fj$	A (g, $\to$I)
4.	$\quad\quad\quad\sim Gj$	A (c, $\sim$E)
5.	$\quad\quad\quad\exists x\sim Gx$	4 $\exists$I
6.	$\quad\quad\quad\bot$	5,2 $\bot$I
7.	$\quad\quad Gj$	4-6 $\sim$E
8.	$\quad Fj \to Gj$	3-7 $\to$I
9.	$\quad\forall x(Fx \to Gx)$	8 $\forall$I
10.	$\quad\bot$	9,1 $\bot$I
11.	$\exists x\sim Gx$	2-10 $\sim$E

The derivation on the left is much to be preferred over the one on the right, where we are caught up in a difficult case of SG5 and then SC3 or SC4. But, after QN, the derivation on the left is straightforward — and would be relatively straightforward even if we missed the uses of Impl and DeM. Observe that, as above, the uses of Impl and DeM help us convert the negated conditional into a conjunction that can be broken into its parts.

Finally, observe that derivations which can be conducted entirely be replacement rules are "reversible." Thus, for a simple case,

(ZZ)

1.	$\sim(A \wedge \sim B)$	A (g, $\leftrightarrow$I)
2.	$\sim A \vee \sim\sim B$	1 DeM
3.	$\sim A \vee B$	2 DN
4.	$A \rightarrow B$	3 Impl
5.	$A \rightarrow B$	A (g, $\leftrightarrow$I)
6.	$\sim A \vee B$	5 Impl
7.	$\sim A \vee \sim\sim B$	6 DN
8.	$\sim(A \wedge \sim B)$	7 DeM
9.	$\sim(A \wedge \sim B) \leftrightarrow (A \rightarrow B)$	1-4,5-8 $\leftrightarrow$I

We set up for $\leftrightarrow$I in the usual way. Then the subderivations work by precisely the same steps, DeM, DN, Impl, but in the reverse order. This is not surprising since replacement rules work in both directions. Notice that reversal does *not* generally work where regular inference rules are involved.

The rules of *ND+* are not a "magic bullet" to make all difficult derivations go away! Rather, with the derived rules, we set aside a certain sort of difficulty that should no longer worry us, so that we are in a position to take on new challenges without becoming overwhelmed by details.

E6.38. Produce derivations to show each of the following.

a. $\sim\exists x(\sim Rx \wedge Sxx),\ Saa \vdash_{ND+} Ra$

b. $\forall x(\sim Axf^1x \vee \exists yBg^1y) \vdash_{ND+} \exists xAf^1xf^1f^1x \rightarrow \exists yBg^1y$

c. $\forall x[(\sim Cxb \vee Hx) \rightarrow Lxx],\ \exists y\sim Lyy \vdash_{ND+} \exists xCxb$

d. $\sim\exists x(Fx \wedge Gx) \vee \exists x\sim Gx,\ \forall yGy \vdash_{ND+} \forall z(Fz \rightarrow \sim Gz)$

e. $\forall xFx,\ \forall zHz \vdash_{ND+} \sim\exists y(\sim Fy \vee \sim Hy)$

*f. $\forall x\forall y\exists zAf^1xyz,\ \forall x\forall y\forall z[Axyz \rightarrow \sim(Cxyz \vee Bzyx)]$
$\vdash_{ND+} \exists x\exists y\sim\forall zBzg^1yf^1g^1x$

g. $\sim\exists x \forall y(Pxy \wedge \sim Qxy) \vdash_{ND+} \forall x \exists y(Pxy \to Qxy)$

h. $\sim\exists y(Ty \vee \exists x \sim Hxy) \vdash_{ND+} \forall x \forall y Hxy \wedge \forall x \sim Tx$

i. $\exists x(Fx \to \exists y \sim Fy) \vdash_{ND+} \sim\forall x Fx$

j. $\vdash_{ND+} \forall x(Ax \to Bx) \vee \exists x Ax$

k. $\vdash_{ND+} \forall x(Fx \vee A) \to (\forall x Fx \vee A)$

l. $\exists x(Fx \leftrightarrow Gx), \forall x[Gx \to (Hx \to Jx)]$
$\vdash_{ND+} \exists x Jx \vee [\forall x Fx \to \exists x(Gx \wedge \sim Hx)]$

m. $\exists x[\sim Bxa \wedge \forall y(Cy \to \sim Gxy)], \forall z[\sim\forall y(Wy \to Gzy) \to Bza]$
$\vdash_{ND+} \forall x(Cx \to \sim Wx)$

*n. $\exists x Fx \to \sim\forall y Gy, \forall x(Kx \to \exists y Jy), \exists y \sim Gy \to \exists x Kx$
$\vdash_{ND+} \sim\exists x Fx \vee \exists y Jy$

o. $\exists z Qz \to \forall w(Lww \to \sim Hw), \exists x Bx \to \forall y(Ay \to Hy)$
$\vdash_{ND+} \exists w(Qw \wedge Bw) \to \forall y(Lyy \to \sim Ay)$

p. $\sim\forall x(\sim Px \vee \sim Hx) \to \forall x[Cx \wedge \forall y(Ly \to Axy)], \exists x[Hx \wedge \forall y(Ly \to Axy)] \to \forall x(Rx \wedge \forall y Bxy) \vdash_{ND+} \sim\forall x \forall y Bxy \to \forall x(\sim Px \vee \sim Hx)$

q. $\vdash_{ND+} (\exists x Ax \to \exists x Bx) \to \exists x(Ax \to Bx)$

r. $\forall x Fx \to A \vdash_{ND+} \exists x(Fx \to A)$

s. $\forall x \exists y(Ax \vee By) \vdash_{ND+} \exists y \forall x(Ax \vee By)$

t. $\forall x Fx \leftrightarrow \sim\exists x \exists y Rxy \vdash_{ND+} \exists x \forall y \forall z(Fx \to \sim Ryz)$

E6.39. Provide derivations to demonstrate BQN for the bounded quantifiers (in the case of $\leq$). That is show T6.68a and T6.68b. Hint: Do not forget your derived rules for the bounded quantifiers.

E6.40. For each of the following, produce a translation into $\mathcal{L}_q$, including interpretation function and formal sentences, and show that the resulting arguments are valid in *ND*.

a. If a first person is taller than a second, then the second is not taller than the first. So nobody is taller than themselves. (An asymmetric relation is irreflexive.)

b. A barber shaves all and only people who do not shave themselves. So there are no barbers.

c. Bob is taller than every other man. If a first person is taller than a second, then the second is not taller than the first. So only Bob is taller than every other man.

d. There is at most one dog, and at least one flea. Any dog is a host for some flea, and any flea has a dog for a host. So there is exactly one dog.

e. Some conception includes god. If one conception includes a thing and another does not, then the greatness of the thing in the first exceeds its greatness in the other. The greatness of no thing in any conception exceeds that of god in a true conception. Therefore, god is included in any true conception.

Hints: Let your universe include conceptions, things in them, along with measures of greatness. Then implement a greatness function $g^2 = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{o}$ is the greatness of m in conception $\mathsf{n}\}$. With an appropriate relation symbol, the greatness of a thing in one conception then exceeds that of a thing in another if something like Eg^2wxg^2yz. This, of course, is a version of Anselm's Ontological Argument. For discussion see, Plantinga, *God, Freedom, and Evil*.

E6.41. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The rules $\forall$I and $\exists$E, including especially restrictions on the rules.

b. The axioms of Q and PA and the way theorems derive from them.

c. The relation between the rules of *ND* and the rules of *ND+*.

Part II

Transition: Reasoning About Logic

Introductory

We have expended a great deal of energy learning to do logic. What we have learned constitutes the complete classical predicate calculus with equality. This is a system of tremendous power including for reasoning in foundations of arithmetic.

But our work itself raises questions. In chapter 4 we used truth trees and tables for an account of the conditions under which sentential formulas are true and arguments are valid. In the quantificational case, though, we were not able to use our graphical methods for a general account of truth and validity — there were simply too many branches, and too many interpretations, for a general account by means of trees. Thus there is an open question about whether and how quantificational validity can be shown.

And once we have introduced our notions of validity, many interesting questions can be asked about how they work: are the arguments that are valid in *AD* the same as the ones that are valid in *ND*? are the arguments that are valid in *ND* the same as the ones that are quantificationally valid? Are the theorems of Q the same as the theorems of PA? are theorems of PA the same as the truths on N the standard interpretation for number theory? Is it possible for a computing device to identify the theorems of the different logical systems?

It is one thing to ask such questions, and perhaps amazing that there are demonstrable answers. We will come to that. However, in this short section we do not attempt answers. Rather, we put ourselves in a position to think about answers by introducing methods for thinking about logic. Thus this part looks both backward and forward: By our methods we plug the hole left from chapter 4: in chapter 7 we accomplish what could not be done with the tables and trees of chapter 4, and are able to demonstrate quantificational validity. At the same time, we lay a foundation to ask and answer core questions about logic.

Chapter 7 begins with our basic method of reasoning from definitions. Chapter 8 introduces mathematical induction. These methods are important not only for results, but for their own sakes, as part of the "package" that comes with mathematical logic.

Chapter 7

Direct Semantic Reasoning

It is the task of this chapter to think about reasoning directly from definitions. Frequently, students who already reason quite skillfully with definitions flounder when asked to do so explicitly, in the style of this chapter.[1] Thus I propose to begin in a restricted context — one with which we are already familiar, using a fairly rigid framework as a guide. Perhaps you first learned to ride a bicycle with training wheels, but eventually learned to ride without them, and so to go faster, and to places other than the wheels would let you go. Similarly, in the end, we will want to apply our methods beyond the restricted context in which we begin, working outside the initial framework. But the framework should give us a good start. In this section, then, I introduce the framework in the context of reasoning for specifically *semantic* notions, and against the background of semantic reasoning we have already done.

In chapter 4 we used truth trees and tables for an account of the conditions under which sentential formulas are true and arguments are valid. In the quantificational case, though, we were not able to use our graphical methods for a general account of truth and validity — there were simply too many branches, and too many interpretations, for a general account by means of trees. For a complete account, we will need to reason more directly from the definitions. But the tables and trees *do* exhibit the semantic definitions. So we can build on what we have already done with them. Our goal will be to move past the tables and trees, and learn to function without

[1] The ability to reason clearly and directly with definitions is important not only here, but also beyond. In philosophy, compare the humorous, but also serious, verb *to chisholm* after Roderick Chisholm, who was a master of the technique — where one proposes a definition; considers a counterexample; modifies to account for the example; considers another counterexample; modifies again; and so forth. As, "He started with definition (d.8) and kept chisholming away at it until he ended up with (d.8′′′′′′′′)" (*The Philosopher's Lexicon*). Such reasoning is impossible to understand apart from explicit attention to consequences of definitions of the sort we have in mind.

them. After some general remarks, we start with the sentential case, and move to the quantificational.

7.1 General

I begin with some considerations about what we are trying to accomplish, and how it is related to what we have done. Consider the following row of a truth table, meant to show that $B \to C \not\models_s \sim B$.

(A)

B	C	$B \to C$ / $\sim B$
T	T	T ***T*** T ***F*** T

Since there is an interpretation on which the premise is true and the conclusion is not, the argument is not sententially valid. Now, what justifies the move from $\mathsf{I}[B] = \mathsf{T}$ and $\mathsf{I}[C] = \mathsf{T}$, to the conclusion that $B \to C$ is T? One might respond, "the truth table." But the truth table, T($\to$) is itself derived from definition ST($\to$). According to ST($\to$), for sentences $\mathcal{P}$ and $\mathcal{Q}$, $\mathsf{I}[(\mathcal{P} \to \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{F}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$ (or both). In this case, $\mathsf{I}[C] = \mathsf{T}$; so $\mathsf{I}[B] = \mathsf{F}$ or $\mathsf{I}[C] = \mathsf{T}$; so the condition from ST($\to$) is met, and $\mathsf{I}[B \to C] = \mathsf{T}$. It may seem odd to move from $\mathsf{I}[C] = \mathsf{T}$; to $\mathsf{I}[B] = \mathsf{F}$ or $\mathsf{I}[C] = \mathsf{T}$, when in fact $\mathsf{I}[B] = \mathsf{T}$; but it is certainly *correct* — just as for $\vee$I in *ND*, the point is merely to make explicit that, in virtue of the fact that $\mathsf{I}[C] = \mathsf{T}$, the interpretation meets the disjunctive condition from ST($\to$). And what justifies the move from $\mathsf{I}[B] = \mathsf{T}$ to the conclusion that $\mathsf{I}[\sim B] = \mathsf{F}$? ST($\sim$). According to ST($\sim$), for any sentence $\mathcal{P}$, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{F}$. In this case, $\mathsf{I}[B] = \mathsf{T}$; and since $\mathsf{I}[B]$ is not F, $\mathsf{I}[\sim B]$ is not T; so $\mathsf{I}[\sim B] = \mathsf{F}$. Similarly, definition SV justifies the conclusion that the argument is not sententially valid. According to SV, $\Gamma \models_s \mathcal{P}$ just in case there is no sentential interpretation I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{P}] = \mathsf{F}$. Since we have produced an I such that $\mathsf{I}[B \to C] = \mathsf{T}$ but $\mathsf{I}[\sim B] = \mathsf{F}$, it follows that $B \to C \not\models_s \sim B$. So the definitions drive the tables.

In chapter 4, we used tables to express these conditions. But we *might* have reasoned directly.

(B) Consider any interpretation I such that $\mathsf{I}[B] = \mathsf{T}$ and $\mathsf{I}[C] = \mathsf{T}$. Since $\mathsf{I}[C] = \mathsf{T}$, $\mathsf{I}[B] = \mathsf{F}$ or $\mathsf{I}[C] = \mathsf{T}$; so by ST($\to$), $\mathsf{I}[B \to C] = \mathsf{T}$. But since $\mathsf{I}[B] = \mathsf{T}$, by ST($\sim$), $\mathsf{I}[\sim B] = \mathsf{F}$. So there is a sentential interpretation I such that $\mathsf{I}[B \to C] = \mathsf{T}$ but $\mathsf{I}[\sim B] = \mathsf{F}$; so by SV, $B \to C \not\models_s \sim B$.

Presumably, all this is "contained" in the one line of the truth table, when we use it to conclude that the argument is not sententially valid.

Similarly, consider the following table, meant to show that $\sim\sim A \models_s \sim A \to A$.

(C)

A	$\sim$	$\sim$	A	/	$\sim$	A	$\to$	A
T	***T***	F	T		F	T	***T***	T
F	***F***	T	F		T	F	***F***	F

Since there is no row where the premise is true and the conclusion is false, the argument is sententially valid. Again, ST($\sim$) and ST($\to$) justify the way you build the table. And SV lets you conclude that the argument is sententially valid. Since no row makes the premise true and the conclusion false, and any sentential interpretation is like some row in its assignment to A, no sentential interpretation makes the premise true and conclusion false; so, by SV, the argument is sententially valid.

Thus the table represents reasoning as follows (omitting the second row). To follow, notice how we simply reason through each "place" in a row, and then about whether the row shows invalidity.

(D) For any sentential interpretation I, either (i) $\mathsf{I}[A] = \mathsf{T}$ or (ii) $\mathsf{I}[A] = \mathsf{F}$. Suppose (i); then $\mathsf{I}[A] = \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim A] = \mathsf{F}$; so by ST($\sim$) again, $\mathsf{I}[\sim\sim A] = \mathsf{T}$. But $\mathsf{I}[A] = \mathsf{T}$, and by ST($\sim$), $\mathsf{I}[\sim A] = \mathsf{F}$; from either of these it follows that $\mathsf{I}[\sim A] = \mathsf{F}$ or $\mathsf{I}[A] = \mathsf{T}$; so by ST($\to$), $\mathsf{I}[\sim A \to A] = \mathsf{T}$. From this either $\mathsf{I}[\sim\sim A] = \mathsf{F}$ or $\mathsf{I}[\sim A \to A] = \mathsf{T}$; so it is not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \to A] = \mathsf{F}$. Suppose (ii); then by related reasoning...it is not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \to A] = \mathsf{F}$. So no interpretation makes it the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \to A] = \mathsf{F}$. So by SV, $\sim\sim A \models_s \sim A \to A$.

Thus we might recapitulate reasoning in the table. Perhaps we typically "whip through" tables without explicitly considering all the definitions involved. But the definitions *are* involved when we complete the table.

Strictly, though, not all of this is necessary for the conclusion that the argument is valid. Thus, for example, in the reasoning at (i), for the conditional there is no need to establish that both $\mathsf{I}[\sim A] = \mathsf{F}$ and that $\mathsf{I}[A] = \mathsf{T}$. From either, it follows that $\mathsf{I}[\sim A] = \mathsf{F}$ or $\mathsf{I}[A] = \mathsf{T}$; and so by ST($\to$) that $\mathsf{I}[\sim A \to A] = \mathsf{T}$. So we might have omitted one or the other. Similarly at (i) there is no need to make the point that $\mathsf{I}[\sim\sim A] = \mathsf{T}$. What matters is that $\mathsf{I}[\sim A \to A] = \mathsf{T}$, so that $\mathsf{I}[\sim\sim A] = \mathsf{F}$ or $\mathsf{I}[\sim A \to A] = \mathsf{T}$, and it is therefore not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \to A] = \mathsf{F}$. So reasoning for the full table might be "shortcut" as follows.

(E) For any sentential interpretation either (i) $\mathsf{I}[A] = \mathsf{T}$ or (ii) $\mathsf{I}[A] = \mathsf{F}$. Suppose (i); then $\mathsf{I}[A] = \mathsf{T}$; so $\mathsf{I}[\sim A] = \mathsf{F}$ or $\mathsf{I}[A] = \mathsf{T}$; so by ST($\to$), $\mathsf{I}[\sim A \to A] = \mathsf{T}$. From this either $\mathsf{I}[\sim\sim A] = \mathsf{F}$ or $\mathsf{I}[\sim A \to A] = \mathsf{T}$; so it is not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \to A] = \mathsf{F}$. Suppose (ii); then $\mathsf{I}[A] = \mathsf{F}$; so by ST($\sim$), $\mathsf{I}[\sim A] = \mathsf{T}$; so by ST($\sim$) again, $\mathsf{I}[\sim\sim A] = \mathsf{F}$; so either $\mathsf{I}[\sim\sim A] = \mathsf{F}$ or $\mathsf{I}[\sim A \to A] = \mathsf{T}$; so it is not the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \to A] = \mathsf{F}$. So no interpretation makes it the case that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \to A] = \mathsf{F}$. So by SV, $\sim\sim A \models_s \sim A \to A$.

This is better. These shortcuts may reflect what you have already done when you realize that, say, a true conclusion eliminates the need to think about the premises on some row of a table. Though the shortcuts make things better, however, the idea of reasoning in this way corresponding to a 4, 8 or more (!) row table remains painful. But there is a way out.

Recall what happens when you apply the short truth-table method from chapter 4 to valid arguments. You start with the assumption that the premises are true and the conclusion is not. If the argument is valid, you reach some conflict so that it is not, in fact, possible to complete the row. Then, as we said on p. 108, you know "in your heart" that the argument is valid. Let us turn this into an official argument form.

(F) Suppose $\sim\sim A \not\models_s \sim A \rightarrow A$; then by SV, there is an I such that $\mathsf{I}[\sim\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A \rightarrow A] = \mathsf{F}$. From the former, by ST($\sim$), $\mathsf{I}[\sim A] = \mathsf{F}$. But from the latter, by ST($\rightarrow$), $\mathsf{I}[\sim A] = \mathsf{T}$ and $\mathsf{I}[A] = \mathsf{F}$; and since $\mathsf{I}[\sim A] = \mathsf{T}$, $\mathsf{I}[\sim A] \neq \mathsf{F}$. This is impossible; reject the assumption: $\sim\sim A \models_s \sim A \rightarrow A$.

This is better still. The assumption that the argument is invalid leads to the conclusion that for some I, $\mathsf{I}[\sim A] = \mathsf{T}$ and $\mathsf{I}[\sim A] = \mathsf{F}$; but a formula is T just in case it is not F, so this is impossible and we reject the assumption. The pattern is like $\sim$I in *ND*. This approach is particularly important insofar as we do not reason individually about each of the possible interpretations. This is nice in the sentential case, when there are too many to reason about conveniently. And in the quantificational case, we will not be *able* to argue individually about each of the possible interpretations. So we need to avoid talking about interpretations one-by-one.

Thus we arrive at two strategies: To show that an argument is invalid, we produce an interpretation, and show by the definitions that it makes the premises true and the conclusion not. That is what we did in (B) above. To show that an argument is valid, we assume the opposite, and show by the definitions that the assumption leads to contradiction. Again, that is what we did just above, at (F).

Before we get to the details, let us consider an important point about what we are trying to do: Our *reasoning* takes place in the metalanguage, based on the definitions — where object-level expressions are *uninterpreted* apart from the definitions. To see this, ask yourself whether a sentence $\mathcal{P}$ conflicts with $\mathcal{P} \mid \mathcal{P}$. "Well," you might respond, "I have never encountered this symbol '$\mid$' before, so I am not in a position to say." But that is the point: whether $\mathcal{P}$ conflicts with $\mathcal{P} \mid \mathcal{P}$ depends entirely on a definition for *stroke* '$\mid$'. As it happens, this symbol is typically read "not-both" as given by what might be a further clause of ST,

ST($\mid$) For any sentences $\mathcal{P}$ and $\mathcal{Q}$, $\mathsf{I}[(\mathcal{P} \mid \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{F}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{F}$ (or both); otherwise $\mathsf{I}[(\mathcal{P} \mid \mathcal{Q})] = \mathsf{F}$.

The resultant table is,

T($\mid$)

$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \mid \mathcal{Q}$
T	T	***F***
T	F	***T***
F	T	***T***
F	F	***T***

$\mathcal{P} \mid \mathcal{Q}$ is false when $\mathcal{P}$ and $\mathcal{Q}$ are both T, and otherwise true. Given this, $\mathcal{P}$ does conflict with $\mathcal{P} \mid \mathcal{P}$. Suppose $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P} \mid \mathcal{P}] = \mathsf{T}$; from the latter, by ST($\mid$), $\mathsf{I}[\mathcal{P}] = \mathsf{F}$ or $\mathsf{I}[\mathcal{P}] = \mathsf{F}$; either way, $\mathsf{I}[\mathcal{P}] = \mathsf{F}$; but this is impossible given our assumption that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$. In fact, $\mathcal{P} \mid \mathcal{P}$ has the same table as $\sim\mathcal{P}$, and $\mathcal{P} \mid (\mathcal{Q} \mid \mathcal{Q})$ the same as $\mathcal{P} \to \mathcal{Q}$.

(G)

$\mathcal{P}$	$\mathcal{P} \mid \mathcal{P}$
T	***F***
F	***T***

$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \mid$	$(\mathcal{Q} \mid \mathcal{Q})$
T	T	***T***	F
T	F	***F***	T
F	T	***T***	F
F	F	***T***	T

From this, we *might* have treated $\sim$ and $\to$, and so $\wedge$, $\vee$ and $\leftrightarrow$, all as abbreviations for expressions whose only operator is $\mid$. At best, however, this leaves official expressions difficult to read. Here is the point that matters: Operators have their significance entirely from the definitions. In this chapter, we make metalinguistic claims *about* object expressions, where these can only be based on the definitions. $\mathcal{P}$ and $\mathcal{P} \mid \mathcal{P}$ do not themselves conflict, apart from the definition which makes $\mathcal{P}$ with $\mathcal{P} \mid \mathcal{P}$ have the consequence that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P}] = \mathsf{F}$. And similarly for operators with which we are more familiar. At every stage, it is the *definitions* which justify conclusions.

7.2 Sentential

With this much said, it remains possible to become confused about details while working with the definitions. It is one thing to be able to follow such reasoning — as I hope you have been able to do — and another to produce it. The idea now is to make use of something at which we are already good, doing derivations, to further structure and guide the way we proceed. The result will be a sort of derivation system for reasoning about definitions. We build up this system in stages.

7.2.1 Truth

Let us begin with some notation. Where the script characters $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D} \ldots$ represent object expressions in the usual way, let the Fraktur characters $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D} \ldots$

represent *metalinguistic* expressions (‘$\mathfrak{A}$’ is the Fraktur ‘A’). Thus $\mathfrak{A}$ might represent an expression of the sort $\mathsf{I}[B] = \mathsf{T}$. Then $\Rightarrow$ and $\Leftrightarrow$ are the metalinguistic conditional and biconditional respectively; $\neg$, $\vartriangle$ and $\triangledown$ represent metalinguistic negation, conjunction, and disjunction. In practice, negation is indicated by the slash ($\nvDash$) as well.

Now consider the following restatement of definition ST. Each clause is given in both a positive and a negative form. For any sentences $\mathcal{P}$ and $\mathcal{Q}$ and interpretation I,

ST (∼)	$\mathsf{I}[\sim\mathcal{P}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T}$	$\mathsf{I}[\sim\mathcal{P}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T}$
(→)	$\mathsf{I}[\mathcal{P} \to \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] = \mathsf{T}$	$\mathsf{I}[\mathcal{P} \to \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$

Given the new symbols, and that a sentence is F iff it is not true, this is a simple restatement of ST. As we develop our formal system, we will treat the metalinguistic biconditionals both as (replacement) rules and as axioms. Thus, for example, it will be legitimate to move by ST(∼) directly from $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ to $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$, moving from right-to-left across the arrow. And similarly in the other direction. Alternatively, it will be appropriate to assert by ST(∼) the entire biconditional, that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T}$. For now, we will mostly use the biconditionals, in the first form, as rules.

To manipulate the definitions, we require some rules. These are like ones you have seen before, only pitched at the metalinguistic level.

com $\quad \mathfrak{A} \triangledown \mathfrak{B} \Leftrightarrow \mathfrak{B} \triangledown \mathfrak{A} \qquad \mathfrak{A} \vartriangle \mathfrak{B} \Leftrightarrow \mathfrak{B} \vartriangle \mathfrak{A}$

idm $\quad \mathfrak{A} \Leftrightarrow \mathfrak{A} \triangledown \mathfrak{A} \qquad \mathfrak{A} \Leftrightarrow \mathfrak{A} \vartriangle \mathfrak{A}$

dem $\quad \neg(\mathfrak{A} \vartriangle \mathfrak{B}) \Leftrightarrow \neg\mathfrak{A} \triangledown \neg\mathfrak{B} \qquad \neg(\mathfrak{A} \triangledown \mathfrak{B}) \Leftrightarrow \neg\mathfrak{A} \vartriangle \neg\mathfrak{B}$

cnj $\quad \dfrac{\mathfrak{A}, \mathfrak{B}}{\mathfrak{A} \vartriangle \mathfrak{B}} \qquad \dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{A}} \qquad \dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{B}}$

dsj $\quad \dfrac{\mathfrak{A}}{\mathfrak{A} \triangledown \mathfrak{B}} \qquad \dfrac{\mathfrak{B}}{\mathfrak{A} \triangledown \mathfrak{B}} \qquad \dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{A}}{\mathfrak{B}} \qquad \dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{B}}{\mathfrak{A}}$

neg $\quad \mathfrak{A} \Leftrightarrow \neg\neg\mathfrak{A}$

$$\begin{array}{|l} \mathfrak{A} \\ \hline \mathfrak{B} \\ \neg\mathfrak{B} \end{array} \qquad \begin{array}{|l} \neg\mathfrak{A} \\ \hline \mathfrak{B} \\ \neg\mathfrak{B} \end{array}$$
$$\neg\mathfrak{A} \qquad\qquad \mathfrak{A}$$

ret $\quad \dfrac{\mathfrak{A}}{\mathfrak{A}}$

Each of these should remind you of rules from *ND* or *ND+*. In practice, we will allow generalized versions of cnj that let us move directly from $\mathfrak{A}_1, \mathfrak{A}_2 \ldots \mathfrak{A}_n$ to $\mathfrak{A}_1 \vartriangle \mathfrak{A}_2 \vartriangle \ldots \vartriangle \mathfrak{A}_n$. Similarly, we will allow applications of dsj and dem that skip officially required applications of neg. Thus, for example, instead of going from $\neg\mathfrak{A} \triangledown \mathfrak{B}$ to $\neg\mathfrak{A} \triangledown \neg\neg\mathfrak{B}$ and then by dem to $\neg(\mathfrak{A} \vartriangle \neg\mathfrak{B})$, we might move by dem

directly from $\neg\mathfrak{A} \bigtriangledown \mathfrak{B}$ to $\neg(\mathfrak{A} \bigtriangleup \neg\mathfrak{B})$. All this should become more clear as we proceed.

With definition ST and these rules, we can begin to reason about consequences of the definition. Suppose we want to show that an interpretation with $\mathsf{I}[A] = \mathsf{I}[B] = \mathsf{T}$ is such that $\mathsf{I}[\sim(A \rightarrow \sim B)] = \mathsf{T}$.

(H)

1.	$\mathsf{I}[A] = \mathsf{T}$	prem
2.	$\mathsf{I}[B] = \mathsf{T}$	prem
3.	$\mathsf{I}[\sim B] \neq \mathsf{T}$	2 ST($\sim$)
4.	$\mathsf{I}[A] = \mathsf{T} \bigtriangleup \mathsf{I}[\sim B] \neq \mathsf{T}$	1,3 cnj
5.	$\mathsf{I}[A \rightarrow \sim B] \neq \mathsf{T}$	4 ST($\rightarrow$)
6.	$\mathsf{I}[\sim(A \rightarrow \sim B)] = \mathsf{T}$	5 ST($\sim$)

We are given that $\mathsf{I}[A] = \mathsf{T}$ and $\mathsf{I}[B] = \mathsf{T}$. From the latter, by ST($\sim$), $\mathsf{I}[\sim B] \neq \mathsf{T}$; so $\mathsf{I}[A] = \mathsf{T}$ and $\mathsf{I}[\sim B] \neq \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{I}[A \rightarrow \sim B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim(A \rightarrow \sim B)] = \mathsf{T}$.

The reasoning on the left is a metalinguistic *derivation* in the sense that every step is either a premise, or justified by a definition or rule. You should be able to follow each step. On the right, we simply "tell the story" of the derivation — mirroring it step-for-step. This latter style is the one we want to develop. As we shall see, it gives us power to go beyond where the formalized derivations will take us. But the derivations serve a purpose. If we can do them, we can *use* them to construct reasoning of the sort we want. Each stage on one side corresponds to one on the other. So the derivations can guide us as we construct our reasoning, and constrain the moves we make. Note: First, on the right, we replace line references with language ("from the latter") meant to serve the same purpose. Second, the metalinguistic symbols, $\Rightarrow$, $\Leftrightarrow$, $\neg$, $\bigtriangleup$, $\bigtriangledown$ are replaced with ordinary language on the right side. Finally, on the right, though we cite every *definition* when we use it, we do not cite the additional *rules* (in this case cnj). In general, as much as possible, you should strive to put the reader (and yourself at a later time) in a position to follow your reasoning — supposing just a basic familiarity with the definitions.

Consider now another example. Suppose we want to show that an interpretation with $\mathsf{I}[B] \neq \mathsf{T}$ is such that $\mathsf{I}[\sim(A \rightarrow \sim B)] \neq \mathsf{T}$.

(I)

1.	$\mathsf{I}[B] \neq \mathsf{T}$	prem
2.	$\mathsf{I}[\sim B] = \mathsf{T}$	1 ST($\sim$)
3.	$\mathsf{I}[A] \neq \mathsf{T} \bigtriangledown \mathsf{I}[\sim B] = \mathsf{T}$	2 dsj
4.	$\mathsf{I}[A \rightarrow \sim B] = \mathsf{T}$	3 ST($\rightarrow$)
5.	$\mathsf{I}[\sim(A \rightarrow \sim B)] \neq \mathsf{T}$	4 ST($\sim$)

We are given that $\mathsf{I}[B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim B] = \mathsf{T}$; so $\mathsf{I}[A] \neq \mathsf{T}$ or $\mathsf{I}[\sim B] = \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{I}[A \rightarrow \sim B] = \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim(A \rightarrow \sim B)] \neq \mathsf{T}$.

Observe that, for a true conditional, on its right-hand side ST($\rightarrow$) *requires* a disjunction sort $\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \bigtriangledown \mathsf{I}[\mathcal{B}] = \mathsf{T}$ — so that (3) yields (4). ST($\rightarrow$) requires a conjunction $\mathsf{I}[\mathcal{P}] = \mathsf{T} \bigtriangleup \mathsf{I}[\mathcal{B}] \neq \mathsf{T}$ just when the conditional is false. Do not get these cases confused, and think that somehow a conjunction of antecedent and consequent yields a

true arrow! Here is another derivation of the same result, this time beginning with the opposite and breaking down to the parts, for an application of neg.

(J)

1.	$\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$	assp
2.	$\mathsf{I}[A \to \sim B] \neq \mathsf{T}$	1 ST($\sim$)
3.	$\mathsf{I}[A] = \mathsf{T} \vartriangle \mathsf{I}[\sim B] \neq \mathsf{T}$	2 ST($\to$)
4.	$\mathsf{I}[\sim B] \neq \mathsf{T}$	3 cnj
5.	$\mathsf{I}[B] = \mathsf{T}$	4 ST($\sim$)
6.	$\mathsf{I}[B] \neq \mathsf{T}$	prem
7.	$\mathsf{I}[\sim(A \to \sim B)] \neq \mathsf{T}$	1-6 neg

Suppose $\mathsf{I}[\sim(A \to \sim B)] = \mathsf{T}$; then from ST($\sim$), $\mathsf{I}[A \to \sim B] \neq \mathsf{T}$; so by ST($\to$), $\mathsf{I}[A] = \mathsf{T}$ and $\mathsf{I}[\sim B] \neq \mathsf{T}$; so $\mathsf{I}[\sim B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[B] = \mathsf{T}$. But we are given that $\mathsf{I}[B] \neq \mathsf{T}$. This is impossible; reject the assumption: $\mathsf{I}[\sim(A \to \sim B)] \neq \mathsf{T}$.

This version takes a couple more lines. But it works as well, and provides a useful illustration of the (neg) rule. As usual, reasonings on the one side mirror that on the other. So we can use the formalized derivation as a guide for the reasoning on the right. Again, we leave out the special metalinguistic symbols. And again we cite all instances of definitions, but not the additional rules.

As you work the exercises that follow, to the extent that you can, it is good to have one line depend on the one before or in the immediate neighborhood, so as to minimize the need for extended references in the written versions. As you work these and other problems, you may find the sentential metalinguistic reference on p. 346 helpful.

E7.1. Suppose $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] \neq \mathsf{T}$ and $\mathsf{I}[C] = \mathsf{T}$. For each of the following, produce a formalized derivation, and then non-formalized reasoning to demonstrate either that it is or is not true on I. Hint: You may find a quick row of the truth table helpful to let you see which you want to show. Also, (e) is much easier than it looks.

a. $\sim B \to C$

*b. $\sim B \to \sim C$

c. $\sim[(A \to \sim B) \to \sim C]$

d. $\sim[A \to (B \to \sim C)]$

e. $\sim A \to [((A \to B) \to C) \to \sim(\sim C \to B)]$

7.2.2 Validity

So far, we have been able to reason about ST and truth. Let us now extend results to validity. For this, we need to augment our formalized system. Let 'S' be a metalinguistic existential quantifier — it asserts the existence of some *object*. For now,

'S' will appear only in contexts asserting the existence of *interpretations*. Thus, for example, $S\mathsf{I}(\mathsf{I}[\mathcal{P}] = \mathsf{T})$ says there is an interpretation I such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$, and $\neg S\mathsf{I}(\mathsf{I}[P] = \mathsf{T})$ says it is not the case that there is an interpretation I such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$. Given this, we can state SV as follows, again in positive and negative forms.

SV $\quad \neg S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \vDash_s \mathcal{Q}$

$\quad S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \nvDash_s \mathcal{Q}$

These should look familiar. An argument is valid when it is *not* the case that there is some interpretation that makes the premises true and the conclusion not. An argument is invalid if there is some interpretation that makes the premises true and the conclusion not.

Again, we need rules to manipulate the new operator. In general, whenever a metalinguistic *term* $\mathfrak{t}$ first appears outside the scope of a metalinguistic quantifier, it is labeled *arbitrary* or *particular*. For the sentential case, terms will always be of the sort I, J..., for *interpretations*, and labeled 'particular' when they first appear apart from the quantifier S. Say $\mathfrak{A}[\mathfrak{t}]$ is some metalinguistic expression in which term $\mathfrak{t}$ appears, and $\mathfrak{A}[\mathfrak{u}]$ is like $\mathfrak{A}[\mathfrak{t}]$ but with free instances of $\mathfrak{t}$ replaced by $\mathfrak{u}$. Perhaps $\mathfrak{A}[\mathfrak{t}]$ is $\mathsf{I}[A] = \mathsf{T}$ and $\mathfrak{A}[\mathfrak{u}]$ is $\mathsf{J}[A] = \mathsf{T}$. Then,

exs $\quad \dfrac{\mathfrak{A}[\mathfrak{u}]}{S\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary or particular $\qquad \dfrac{S\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ particular and new

As an instance of the left-hand "introduction" rule, we might move from $\mathsf{J}[A] = \mathsf{T}$, for a J labeled either arbitrary or particular, to $S\mathsf{I}(\mathsf{I}[A] = \mathsf{T})$. If interpretation J is such that $\mathsf{J}[A] = \mathsf{T}$, then there is *some* interpretation I such that $\mathsf{I}[A] = \mathsf{T}$. For the other "exploitation" rule, we may move from $S\mathsf{I}(\mathsf{I}[A] = \mathsf{T})$ to the result that $\mathsf{J}[A] = \mathsf{T}$ so long as J is identified as *particular* and is new to the derivation, in the sense required for $\exists$E in chapter 6. In particular, it must be that the term does not so-far appear outside the scope of a metalinguistic quantifier, and does not appear free in the final result of the derivation. Given that some I is such that $\mathsf{I}[A] = \mathsf{T}$, we set up J as a particular interpretation for which it is so.[2]

In addition, it will be helpful to allow a rule which lets us make assertions by *inspection* about already given interpretations — and we will limit justifications by

[2]Observe that, insofar as it is quantified, term I may itself be new in the sense that it does not so far appear outside the scope of a quantifier. Thus we may be justified in moving from $S\mathsf{I}(\mathsf{I}[A] = \mathsf{T})$ to $\mathsf{I}[A] = \mathsf{T}$, with I particular. However, as a matter of style, we will typically switch terms upon application of the exs rule.

(ins) just to assertions about interpretations (and, later, variable assignments). Thus, for example, in the context of an interpretation I on which $\mathsf{I}[A] = \mathsf{T}$, we might allow,

n. $\mathsf{I}[A] = \mathsf{T}$ ins (I particular)

as a line of one of our derivations. In this case, I is a *name* of the interpretation, and listed as particular on first use.

Now suppose we want to show that $(B \rightarrow \sim D), \sim B \not\vDash_s D$. Recall that our strategy for showing that an argument is invalid is to *produce* an interpretation, and show that it makes the premises true and the conclusion not. So consider an interpretation J such that $\mathsf{J}[B] \neq \mathsf{T}$ and $\mathsf{J}[D] \neq \mathsf{T}$.

(K)

1.	$\mathsf{J}[B] \neq \mathsf{T}$	ins (J particular)
2.	$\mathsf{J}[B] \neq \mathsf{T} \triangledown \mathsf{J}[\sim D] = \mathsf{T}$	1 dsj
3.	$\mathsf{J}[B \rightarrow \sim D] = \mathsf{T}$	2 ST($\rightarrow$)
4.	$\mathsf{J}[\sim B] = \mathsf{T}$	1 ST($\sim$)
5.	$\mathsf{J}[D] \neq \mathsf{T}$	ins
6.	$\mathsf{J}[B \rightarrow \sim D] = \mathsf{T} \vartriangle \mathsf{J}[\sim B] = \mathsf{T} \vartriangle \mathsf{J}[D] \neq \mathsf{T}$	3,4,5 cnj
7.	$S\mathsf{I}(\mathsf{I}[B \rightarrow \sim D] = \mathsf{T} \vartriangle \mathsf{I}[\sim B] = \mathsf{T} \vartriangle \mathsf{I}[D] \neq \mathsf{T})$	6 exs
8.	$B \rightarrow \sim D, \sim B \not\vDash_s D$	7 SV

(1) and (5) are by inspection of the interpretation J, where an individual name is always labeled "particular" when it first appears. At (6) we have a conclusion about interpretation J, and at (7) we generalize to the existential, for an application of SV at (8). Here is the corresponding informal reasoning.

> $\mathsf{J}[B] \neq \mathsf{T}$; so either $\mathsf{J}[B] \neq \mathsf{T}$ or $\mathsf{J}[\sim D] = \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{J}[B \rightarrow \sim D] = \mathsf{T}$. But since $\mathsf{J}[B] \neq \mathsf{T}$, by ST($\sim$), $\mathsf{J}[\sim B] = \mathsf{T}$. And $\mathsf{J}[D] \neq \mathsf{T}$. So $\mathsf{J}[B \rightarrow \sim D] = \mathsf{T}$ and $\mathsf{J}[\sim B] = \mathsf{T}$ but $\mathsf{J}[D] \neq \mathsf{T}$. So there is an interpretation I such that $\mathsf{I}[B \rightarrow \sim D] = \mathsf{T}$ and $\mathsf{I}[\sim B] = \mathsf{T}$ but $\mathsf{I}[D] \neq \mathsf{T}$. So by SV, $(B \rightarrow \sim D), \sim B \not\vDash_s D$

It should be clear that this reasoning reflects that of the derivation. The derivation thus constrains the steps we make, and guides us to our goal. We show the argument is invalid by showing that there exists an interpretation on which the premises are true and the conclusion is not.

Say we want to show that $\sim(A \rightarrow B) \vDash_s A$. To show that an argument is valid, our idea has been to assume otherwise, and show that the assumption leads to contradiction. So we might reason as follows.

	1.	$\sim(A \to B) \not\models_s A$	assp
	2.	$S\mathsf{I}(\mathsf{I}[\sim(A \to B)] = \mathsf{T} \vartriangle \mathsf{I}[A] \neq \mathsf{T})$	1 SV
	3.	$\mathsf{J}[\sim(A \to B)] = \mathsf{T} \vartriangle \mathsf{J}[A] \neq \mathsf{T}$	2 exs (J particular)
	4.	$\mathsf{J}[\sim(A \to B)] = \mathsf{T}$	3 cnj
(L)	5.	$\mathsf{J}[A \to B] \neq \mathsf{T}$	4 ST($\sim$)
	6.	$\mathsf{J}[A] = \mathsf{T} \vartriangle \mathsf{J}[B] \neq \mathsf{T}$	5 ST($\to$)
	7.	$\mathsf{J}[A] = \mathsf{T}$	6 cnj
	8.	$\mathsf{J}[A] \neq \mathsf{T}$	3 cnj
	9.	$\sim(A \to B) \models_s A$	1-8 neg

Suppose $\sim(A \to B) \not\models_s A$; then by SV there is some I such that $\mathsf{I}[\sim(A \to B)] = \mathsf{T}$ and $\mathsf{I}[A] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[\sim(A \to B)] = \mathsf{T}$ and $\mathsf{J}[A] \neq \mathsf{T}$. From the former, by ST($\sim$), $\mathsf{J}[A \to B] \neq \mathsf{T}$; so by ST($\to$), $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$. So both $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[A] \neq \mathsf{T}$. This is impossible; reject the assumption: $\sim(A \to B) \models_s A$.

At (2) we have the result that there is some interpretation on which the premise is true and the conclusion is not. At (3), we set up to reason about a particular J for which this is so. J does not so-far appear in the derivation, and does not appear in the goal at (9). So we instantiate to it. This puts us in a position to reason by ST. The pattern is typical. Given that the assumption leads to contradiction, we are justified in rejecting the assumption, and thus conclude that the argument is valid. It is important that we show the argument is valid, without reasoning individually about every possible interpretation of the basic sentences!

Notice that we can also reason generally about *forms*. Here is a case of that sort.

T7.4s. $\models_s (\sim\mathcal{Q} \to \sim\mathcal{P}) \to [(\sim\mathcal{Q} \to \mathcal{P}) \to \mathcal{Q}]$

1.	$\nvDash_s (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$	assp
2.	$S\mathsf{I}(\mathsf{I}[(\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})] \neq \mathsf{T})$	1 SV
3.	$\mathsf{J}[(\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[\sim\mathcal{Q} \rightarrow \sim\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{J}[(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}] \neq \mathsf{T}$	3 ST($\rightarrow$)
5.	$\mathsf{J}[(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}] \neq \mathsf{T}$	4 cnj
6.	$\mathsf{J}[\sim\mathcal{Q} \rightarrow \mathcal{P}] = \mathsf{T} \vartriangle \mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$	5 ST($\rightarrow$)
7.	$\mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$	6 cnj
8.	$\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$	7 SF($\sim$)
9.	$\mathsf{J}[\sim\mathcal{Q} \rightarrow \mathcal{P}] = \mathsf{T}$	6 cnj
10.	$\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T} \triangledown \mathsf{J}[\mathcal{P}] = \mathsf{T}$	9 ST($\rightarrow$)
11.	$\mathsf{J}[\mathcal{P}] = \mathsf{T}$	8,10 dsj
12.	$\mathsf{J}[\sim\mathcal{Q} \rightarrow \sim\mathcal{P}] = \mathsf{T}$	4 cnj
13.	$\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T} \triangledown \mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$	12 ST($\rightarrow$)
14.	$\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$	8,13 dsj
15.	$\mathsf{J}[\mathcal{P}] \neq \mathsf{T}$	14 ST($\sim$)
16.	$\vDash_s (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$	1-15 neg

Suppose $\nvDash_s (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$; then by SV there is some I such that $\mathsf{I}[(\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[(\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})] \neq \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{J}[\sim\mathcal{Q} \rightarrow \sim\mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}] \neq \mathsf{T}$; from the latter, by ST($\rightarrow$), $\mathsf{J}[\sim\mathcal{Q} \rightarrow \mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$; from the latter of these, by ST($\sim$), $\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$. Since $\mathsf{J}[\sim\mathcal{Q} \rightarrow \mathcal{P}] = \mathsf{T}$, by ST($\rightarrow$), $\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T}$ or $\mathsf{J}[\mathcal{P}] = \mathsf{T}$; but $\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$, so $\mathsf{J}[\mathcal{P}] = \mathsf{T}$. Since $\mathsf{J}[\sim\mathcal{Q} \rightarrow \sim\mathcal{P}] = \mathsf{T}$, by ST($\rightarrow$), $\mathsf{J}[\sim\mathcal{Q}] \neq \mathsf{T}$ or $\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$; but $\mathsf{J}[\sim\mathcal{Q}] = \mathsf{T}$, so $\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[\mathcal{P}] \neq \mathsf{T}$. This is impossible; reject the assumption: $\vDash_s (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$.

Observe that the steps represented by (11) and (14) are not by cnj but by the dsj rule with $\mathfrak{A} \triangledown \mathfrak{B}$ and $\neg\mathfrak{A}$ for the result that $\mathfrak{B}$.[3] Observe also that contradictions are obtained at the *metalinguistic* level. Thus $\mathsf{J}[\mathcal{P}] = \mathsf{T}$ at (11) does not contradict $\mathsf{J}[\sim\mathcal{P}] = \mathsf{T}$ at (14). Of course, it is a short step to the result that $\mathsf{J}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{P}] \neq \mathsf{T}$ which do contradict. As a general point of strategy, it is much easier to manage a negated conditional than an unnegated one — for the negated conditional yields a conjunctive result, and the unnegated a disjunctive. Thus we begin above with the negated conditionals, and *use* the results to set up applications of dsj. This is typical.

There is nothing special about reasoning with forms. Thus similarly we can show,

T7.1s. $\mathcal{P}, \mathcal{P} \rightarrow \mathcal{Q} \vDash_s \mathcal{Q}$

[3] Or, rather, we have $\neg\mathfrak{A} \triangledown \mathfrak{B}$ and $\mathfrak{A}$ — and thus skip application of neg to obtain the proper $\neg\neg\mathfrak{A}$ for this application of dsj.

T7.2s. $\vDash_s \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$

T7.3s. $\vDash_s (\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})) \rightarrow ((\mathcal{O} \rightarrow \mathcal{P}) \rightarrow (\mathcal{O} \rightarrow \mathcal{Q}))$

T7.1s - T7.4s should remind you of the axioms and rule for the sentential part of *AD* from chapter 3. These results (or, rather, analogues for the quantificational case) play an important role for things to come.

These derivations are structurally much simpler than ones you have seen before from *ND*. The challenge is accommodating new notation with the different mix of rules. Again, to show that an argument is invalid, produce an interpretation; then use it for a demonstration that there exists an interpretation that makes premises true and the conclusion not. To show that an argument is valid, suppose otherwise; then demonstrate that your assumption leads to contradiction. The derivations then provide the pattern for your informal reasoning.

E7.2. Produce a formalized derivation, and then informal reasoning to demonstrate each of the following. To show invalidity, you will have to *produce* an interpretation to which your argument refers.

*a. $A \rightarrow B, \sim A \nvDash_s \sim B$

*b. $A \rightarrow B, \sim B \vDash_s \sim A$

c. $A \rightarrow B, B \rightarrow C, C \rightarrow D \vDash_s A \rightarrow D$

d. $A \rightarrow B, B \rightarrow \sim A \vDash_s \sim A$

e. $A \rightarrow B, \sim A \rightarrow \sim B \nvDash_s \sim(A \rightarrow \sim B)$

f. $(\sim A \rightarrow B) \rightarrow A \vDash_s \sim A \rightarrow \sim B$

g. $\sim A \rightarrow \sim B, B \vDash_s \sim(B \rightarrow \sim A)$

h. $A \rightarrow B, \sim B \rightarrow A \nvDash_s A \rightarrow \sim B$

i. $\nvDash_s [(A \rightarrow B) \rightarrow (A \rightarrow C)] \rightarrow [(A \rightarrow B) \rightarrow C]$

j. $\vDash_s (A \rightarrow B) \rightarrow [(B \rightarrow \sim C) \rightarrow (C \rightarrow \sim A)]$

E7.3. Provide demonstrations for T7.1s - T7.3s in the informal style. Hint: you may or may not find that truth tables, or formalized derivations, would be helpful as a guide.

7.2.3 Derived Rules

Finally, for this section on sentential forms, we expand the range of our results by means of some rules for $\Rightarrow$ and $\Leftrightarrow$.

cnd $\quad \dfrac{\mathfrak{A} \Rightarrow \mathfrak{B},\ \mathfrak{A}}{\mathfrak{B}} \qquad \begin{array}{l} \left|\begin{array}{l}\mathfrak{A}\\ \hline \\ \mathfrak{B}\end{array}\right. \\ \mathfrak{A} \Rightarrow \mathfrak{B}\end{array} \qquad \dfrac{\mathfrak{A} \Rightarrow \mathfrak{B},\ \mathfrak{B} \Rightarrow \mathfrak{C}}{\mathfrak{A} \Rightarrow \mathfrak{C}}$

bcnd $\quad \dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B},\ \mathfrak{A}}{\mathfrak{B}} \qquad \dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B},\ \mathfrak{B}}{\mathfrak{A}} \qquad \dfrac{\mathfrak{A} \Rightarrow \mathfrak{B},\ \mathfrak{B} \Rightarrow \mathfrak{A}}{\mathfrak{A} \Leftrightarrow \mathfrak{B}} \qquad \dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B},\ \mathfrak{B} \Leftrightarrow \mathfrak{C}}{\mathfrak{A} \Leftrightarrow \mathfrak{C}}$

We will also allow versions of bcnd which move from, say, $\mathfrak{A} \Leftrightarrow \mathfrak{B}$ and $\neg\mathfrak{A}$, to $\neg\mathfrak{B}$ (like NB from *ND+*). And we will allow generalized versions of these rules moving directly from, say, $\mathfrak{A} \Rightarrow \mathfrak{B}$, $\mathfrak{B} \Rightarrow \mathfrak{C}$, and $\mathfrak{C} \Rightarrow \mathfrak{D}$ to $\mathfrak{A} \Rightarrow \mathfrak{D}$; and similarly, from $\mathfrak{A} \Leftrightarrow \mathfrak{B}$, $\mathfrak{B} \Leftrightarrow \mathfrak{C}$, and $\mathfrak{C} \Leftrightarrow \mathfrak{D}$ to $\mathfrak{A} \Leftrightarrow \mathfrak{D}$. In this last case, the natural informal description is, $\mathfrak{A}$ iff $\mathfrak{B}$; $\mathfrak{B}$ iff $\mathfrak{C}$; $\mathfrak{C}$ iff $\mathfrak{D}$; so $\mathfrak{A}$ iff $\mathfrak{D}$. In real cases, however, repetition of terms can be awkward and get in the way of reading. In practice, then, the pattern collapses to, $\mathfrak{A}$ iff $\mathfrak{B}$; iff $\mathfrak{C}$; iff $\mathfrak{D}$; so $\mathfrak{A}$ iff $\mathfrak{D}$ — where this is understood as in the official version.

Also, when demonstrating that $\mathfrak{A} \Rightarrow \mathfrak{B}$, in many cases, it is helpful to get $\mathfrak{B}$ by neg; officially, the pattern is as on the left,

$$\begin{array}{l} \left|\begin{array}{l}\mathfrak{A}\\ \hline \left|\begin{array}{l}\neg\mathfrak{B}\\ \hline \mathfrak{C}\\ \neg\mathfrak{C}\end{array}\right. \\ \mathfrak{B}\end{array}\right. \\ \mathfrak{A} \Rightarrow \mathfrak{B}\end{array}$$

But the result is automatic once we derive a contradiction from $\mathfrak{A}$ and $\neg\mathfrak{B}$; so, in practice, this pattern collapses into:

$$\begin{array}{l} \left|\begin{array}{l}\mathfrak{A} \vartriangle \neg\mathfrak{B}\\ \hline \mathfrak{C}\\ \neg\mathfrak{C}\end{array}\right. \\ \mathfrak{A} \Rightarrow \mathfrak{B}\end{array}$$

So to demonstrate a conditional, it is enough to derive a contradiction from the antecedent and negation of the consequent. Let us also include among our definitions, (abv) for unpacking abbreviations. This is to be understood as justifying any biconditional $\mathfrak{A} \Leftrightarrow \mathfrak{A}'$ where $\mathfrak{A}'$ abbreviates $\mathfrak{A}$. Such a biconditional can be used as either an axiom or a rule.

We are now in a position to produce derived clauses for ST. In table form, we have already seen derived forms for ST from chapter 4. But we did not then have the official means to extend the definition.

ST′ $\quad (\wedge)\ \ \mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}$

$$\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$$

$$(\vee)\quad \mathsf{I}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] = \mathsf{T}$$

$$\mathsf{I}[\mathcal{P} \vee \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$$

$$(\leftrightarrow)\quad \mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] = \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}) \triangledown (\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T})$$

$$\mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \triangledown (\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T})$$

Again, you should recognize the derived clauses based on what you already know from truth tables.

First, consider the positive form for ST′(∧). We reason about the arbitrary interpretation. The demonstration begins by abv, and strings together biconditionals to reach the final result.

(M)

1.	$\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\sim(\mathcal{P} \to \sim\mathcal{Q})] = \mathsf{T}$	abv (I arbitrary)
2.	$\mathsf{I}[\sim(\mathcal{P} \to \sim\mathcal{Q})] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P} \to \sim\mathcal{Q}] \neq \mathsf{T}$	ST(∼)
3.	$\mathsf{I}[\mathcal{P} \to \sim\mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T}$	ST(→)
4.	$\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}$	ST(∼)
5.	$\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}$	1,2,3,4 bcnd

This derivation puts together a string of biconditionals of the form $\mathfrak{A} \Leftrightarrow \mathfrak{B}$, $\mathfrak{B} \Leftrightarrow \mathfrak{C}$, $\mathfrak{C} \Leftrightarrow \mathfrak{D}$, $\mathfrak{D} \Leftrightarrow \mathfrak{E}$; the conclusion follows by bcnd. Notice that we use the abbreviation and first two definitions as axioms, to state the biconditonals. Technically, (4) results from an implicit $\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T}$ with ST(∼) as a replacement rule, substituting $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$ for $\mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T}$ on the right-hand side. In the “collapsed” biconditional form, the result is as follows.

> By abv, $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\sim(\mathcal{P} \to \sim\mathcal{Q})] = \mathsf{T}$; by ST(∼), iff $\mathsf{I}[\mathcal{P} \to \sim\mathcal{Q}] \neq \mathsf{T}$; by ST(→), iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathcal{Q}] \neq \mathsf{T}$; by ST(∼), iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$. So $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$.

In this abbreviated form, each stage implies the next from start to finish. But similarly, each stage implies the one before from finish to start. So one might think of it as demonstrating conditionals in both directions all at once for eventual application of bcnd. Because we have just shown a biconditional, it follows immediately that $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T}$ just in case the right hand side fails — just in case one of $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$. However, we can also make the point directly.

> By abv, $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T}$ iff $\mathsf{I}[\sim(\mathcal{P} \to \sim\mathcal{Q})] \neq \mathsf{T}$; by ST(∼), iff $\mathsf{I}[\mathcal{P} \to \sim\mathcal{Q}] = \mathsf{T}$; by ST(→), iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\sim\mathcal{Q}] = \mathsf{T}$; by ST(∼), iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$. So $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$.

Reasoning for ST$'(\vee)$ is similar. For ST$'(\leftrightarrow)$ it will be helpful to introduce, as a derived rule, a sort of distribution principle.

dst $\quad [(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \Leftrightarrow [(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$

To show this, our basic idea will be to obtain the conditional going in both directions, and then apply bcnd. Here is the argument from left-to-right.

1.	$[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \vartriangle \neg[(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$	assp
2.	$\neg[(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$	1 cnj
3.	$(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})$	1 cnj
4.	$\neg\mathfrak{A} \triangledown \mathfrak{B}$	3 cnj
5.	$\neg\mathfrak{B} \triangledown \mathfrak{A}$	3 cnj
6.	$\neg(\mathfrak{A} \vartriangle \mathfrak{B}) \vartriangle \neg(\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	2 dem
7.	$\neg(\mathfrak{A} \vartriangle \mathfrak{B})$	6 cnj
8.	$\neg\mathfrak{A} \triangledown \neg\mathfrak{B}$	7 dem
9.	$\quad \mathfrak{A}$	assp
10.	$\quad \mathfrak{B}$	4,9 dsj
11.	$\quad \neg\mathfrak{B}$	8,9 dsj
12.	$\neg\mathfrak{A}$	9-11 neg
13.	$\neg\mathfrak{B}$	5,12 dsj
14.	$\neg(\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	6 cnj
15.	$\mathfrak{A} \triangledown \mathfrak{B}$	14 dem
16.	$\mathfrak{B}$	12,15 dsj
17.	$[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \Rightarrow [(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$	1-16 cnd

The conditional is demonstrated in the "collapsed" form, where we assume the antecedent with the negation of the consequent, and go for a contradiction. Note the little subderivation at (9) - (11); often the way to make headway with metalinguistic disjunction is to assume the negation of one side. This can feed into dsj and neg. Demonstration of the conditional in the other direction is left as an exercise. Given dst, you should be able to demonstrate ST$(\leftrightarrow)$, also in the collapsed biconditional style. You will begin by observing by abv that $\mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\sim((\mathcal{P} \to \mathcal{Q}) \to \sim(\mathcal{Q} \to \mathcal{P}))] = \mathsf{T}$; by neg iff .... The negative side is relatively straightforward, and does not require dst.

Having established the derived clauses for ST$'$, we can use them directly in our reasoning. Thus, for example, let us show that $B \vee (A \wedge \sim C), (C \to A) \leftrightarrow B \not\models_s \sim(A \wedge C)$. For this, consider an interpretation J such that $\mathsf{J}[A] = \mathsf{J}[B] = \mathsf{J}[C] = \mathsf{T}$.

Metalinguistic Quick Reference (sentential)

DEFINITIONS:

ST ($\sim$) $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ $\quad$ $\mathsf{I}[\sim\mathcal{P}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T}$

($\rightarrow$) $\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] = \mathsf{T}$ $\quad$ $\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$

($\mid$) $\mathsf{I}[\mathcal{P} \mid \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$ $\quad$ $\mathsf{I}[\mathcal{P} \mid \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}$

ST′ ($\wedge$) $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}$

$\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$.

($\vee$) $\mathsf{I}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] = \mathsf{T} \triangledown \mathsf{I}[\mathcal{Q}] = \mathsf{T}$

$\mathsf{I}[\mathcal{P} \vee \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$.

($\leftrightarrow$) $\mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] = \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T}) \triangledown (\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T})$

$\mathsf{I}[\mathcal{P} \leftrightarrow \mathcal{Q}] \neq \mathsf{T} \Leftrightarrow (\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \triangledown (\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] = \mathsf{T})$.

SV $\neg S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \vDash_s \mathcal{Q}$

$S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \nvDash_s \mathcal{Q}$

abv *Abbreviation* allows $\mathfrak{A} \Leftrightarrow \mathfrak{A}'$ where $\mathfrak{A}'$ abbreviates $\mathfrak{A}$.

RULES:

com $\mathfrak{A} \triangledown \mathfrak{B} \Leftrightarrow \mathfrak{B} \triangledown \mathfrak{A}$ $\quad$ $\mathfrak{A} \vartriangle \mathfrak{B} \Leftrightarrow \mathfrak{B} \vartriangle \mathfrak{A}$

idm $\mathfrak{A} \Leftrightarrow \mathfrak{A} \triangledown \mathfrak{A}$ $\quad$ $\mathfrak{A} \Leftrightarrow \mathfrak{A} \vartriangle \mathfrak{A}$

dem $\neg(\mathfrak{A} \vartriangle \mathfrak{B}) \Leftrightarrow \neg\mathfrak{A} \triangledown \neg\mathfrak{B}$ $\quad$ $\neg(\mathfrak{A} \triangledown \mathfrak{B}) \Leftrightarrow \neg\mathfrak{A} \vartriangle \neg\mathfrak{B}$

cnj $\dfrac{\mathfrak{A}, \mathfrak{B}}{\mathfrak{A} \vartriangle \mathfrak{B}}$ $\quad$ $\dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{A}}$ $\quad$ $\dfrac{\mathfrak{A} \vartriangle \mathfrak{B}}{\mathfrak{B}}$

dsj $\dfrac{\mathfrak{A}}{\mathfrak{A} \triangledown \mathfrak{B}}$ $\quad$ $\dfrac{\mathfrak{B}}{\mathfrak{A} \triangledown \mathfrak{B}}$ $\quad$ $\dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{A}}{\mathfrak{B}}$ $\quad$ $\dfrac{\mathfrak{A} \triangledown \mathfrak{B}, \neg\mathfrak{B}}{\mathfrak{A}}$

neg $\mathfrak{A} \Leftrightarrow \neg\neg\mathfrak{A}$ $\quad$ $\begin{array}{l} \left| \begin{array}{l} \underline{\mathfrak{A}} \\ \mathfrak{B} \\ \neg\mathfrak{B} \end{array} \right. \\ \neg\mathfrak{A} \end{array}$ $\quad$ $\begin{array}{l} \left| \begin{array}{l} \underline{\neg\mathfrak{A}} \\ \mathfrak{B} \\ \neg\mathfrak{B} \end{array} \right. \\ \mathfrak{A} \end{array}$ $\quad$ ret $\dfrac{\mathfrak{A}}{\mathfrak{A}}$

exs $\dfrac{\mathfrak{A}[\mathfrak{u}]}{S\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary or particular $\quad$ $\dfrac{S\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ particular and new

cnd $\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{A}}{\mathfrak{B}}$ $\quad$ $\begin{array}{l} \left| \begin{array}{l} \underline{\mathfrak{A}} \\ \mathfrak{B} \end{array} \right. \\ \mathfrak{A} \Rightarrow \mathfrak{B} \end{array}$ $\quad$ $\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{B} \Rightarrow \mathfrak{C}}{\mathfrak{A} \Rightarrow \mathfrak{C}}$ $\quad$ $\begin{array}{l} \left| \begin{array}{l} \underline{\mathfrak{A} \vartriangle \neg\mathfrak{B}} \\ \mathfrak{C} \\ \neg\mathfrak{C} \end{array} \right. \\ \mathfrak{A} \Rightarrow \mathfrak{B} \end{array}$

bcnd $\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{A}}{\mathfrak{B}}$ $\quad$ $\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{B}}{\mathfrak{A}}$ $\quad$ $\dfrac{\mathfrak{A} \Rightarrow \mathfrak{B}, \mathfrak{B} \Rightarrow \mathfrak{A}}{\mathfrak{A} \Leftrightarrow \mathfrak{B}}$ $\quad$ $\dfrac{\mathfrak{A} \Leftrightarrow \mathfrak{B}, \mathfrak{B} \Leftrightarrow \mathfrak{C}}{\mathfrak{A} \Leftrightarrow \mathfrak{C}}$

dst $[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})] \Leftrightarrow [(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})]$

ins *Inspection* allows assertions about interpretations and variable assignments.

(N)

1.	$\mathsf{J}[A] = \mathsf{T}$	ins (J particular)
2.	$\mathsf{J}[C] = \mathsf{T}$	ins
3.	$\mathsf{J}[A] = \mathsf{T} \vartriangle \mathsf{J}[C] = \mathsf{T}$	1,2 cnj
4.	$\mathsf{J}[A \wedge C] = \mathsf{T}$	3 ST$'$($\wedge$)
5.	$\mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$	4 ST($\sim$)
6.	$\mathsf{J}[B] = \mathsf{T}$	ins
7.	$\mathsf{J}[B] = \mathsf{T} \triangledown \mathsf{J}[A \wedge \sim C] = \mathsf{T}$	6 dsj
8.	$\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T}$	7 ST$'$($\vee$)
9.	$\mathsf{J}[C] \neq \mathsf{T} \triangledown \mathsf{J}[A] = \mathsf{T}$	1, dsj
10.	$\mathsf{J}[C \to A] = \mathsf{T}$	9 ST($\to$)
11.	$\mathsf{J}[C \to A] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T}$	10,6 cnj
12.	$(\mathsf{J}[C \to A] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T}) \triangledown (\mathsf{J}[C \to A] \neq \mathsf{T} \vartriangle \mathsf{J}[B] \neq \mathsf{T})$	11 dsj
13.	$\mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T}$	12, ST$'$($\leftrightarrow$)
14.	$\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T} \vartriangle \mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$	8,13,5 cnj
15.	$S\mathsf{I}[\mathsf{I}[B \vee (A \wedge \sim C)] = \mathsf{T} \vartriangle \mathsf{I}[(C \to A) \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{I}[\sim(A \wedge C)] \neq \mathsf{T}]$	14 exs
16.	$B \vee (A \wedge \sim C), (C \to A) \leftrightarrow B \not\models_s \sim(A \wedge C)$	15 SV

Since $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[C] = \mathsf{T}$, by ST$'$($\wedge$), $\mathsf{J}[A \wedge C] = \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$. Since $\mathsf{J}[B] = \mathsf{T}$, either $\mathsf{J}[B] = \mathsf{T}$ or $\mathsf{J}[A \wedge \sim C] = \mathsf{T}$; so by ST$'$($\vee$), $\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T}$. Since $\mathsf{J}[A] = \mathsf{T}$, either $\mathsf{J}[C] \neq \mathsf{T}$ or $\mathsf{J}[A] = \mathsf{T}$; so by ST($\to$), $\mathsf{J}[C \to A] = \mathsf{T}$; so both $\mathsf{J}[C \to A] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$; so either both $\mathsf{J}[C \to A] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$ or both $\mathsf{J}[C \to A] \neq \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$; so by ST$'$($\leftrightarrow$), $\mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T}$. So $\mathsf{J}[B \vee (A \wedge \sim C)] = \mathsf{T}$ and $\mathsf{J}[(C \to A) \leftrightarrow B] = \mathsf{T}$ but $\mathsf{J}[\sim(A \wedge C)] \neq \mathsf{T}$; so there exists an interpretation I such that $\mathsf{I}[B \vee (A \wedge \sim C)] = \mathsf{T}$ and $\mathsf{I}[(C \to A) \leftrightarrow B] = \mathsf{T}$ but $\mathsf{I}[\sim(A \wedge C)] \neq \mathsf{T}$; so by SV, $B \vee (A \wedge \sim C)$, $(C \to A) \leftrightarrow B \not\models_s \sim(A \wedge C)$.

Similarly we can show that $A \to (B \vee C), C \leftrightarrow B, \sim C \models_s \sim A$. As usual, our strategy is to assume otherwise, and go for contradiction.

	1.	$A \to (B \vee C), C \leftrightarrow B, {\sim}C \nvDash_s {\sim}A$	assp
	2.	$S\mathsf{I}(\mathsf{I}[A \to (B \vee C)] = \mathsf{T} \vartriangle \mathsf{I}[C \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{I}[{\sim}C] = \mathsf{T} \vartriangle \mathsf{I}[{\sim}A] \neq \mathsf{T})$	1 SV
	3.	$\mathsf{J}[A \to (B \vee C)] = \mathsf{T} \vartriangle \mathsf{J}[C \leftrightarrow B] = \mathsf{T} \vartriangle \mathsf{J}[{\sim}C] = \mathsf{T} \vartriangle \mathsf{J}[{\sim}A] \neq \mathsf{T}$	2 exs (J particular)
	4.	$\mathsf{J}[{\sim}C] = \mathsf{T}$	3 cnj
	5.	$\mathsf{J}[C] \neq \mathsf{T}$	4 ST($\sim$)
	6.	$\mathsf{J}[C] \neq \mathsf{T} \triangledown \mathsf{J}[B] \neq \mathsf{T}$	5 dsj
	7.	$\neg(\mathsf{J}[C] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T})$	6 dem
	8.	$\mathsf{J}[C \leftrightarrow B] = \mathsf{T}$	3 cnj
	9.	$(\mathsf{J}[C] = \mathsf{T} \vartriangle \mathsf{J}[B] = \mathsf{T}) \triangledown (\mathsf{J}[C] \neq \mathsf{T} \vartriangle \mathsf{J}[B] \neq \mathsf{T})$	8 ST$'$($\leftrightarrow$)
(O)	10.	$\mathsf{J}[C] \neq \mathsf{T} \vartriangle \mathsf{J}[B] \neq \mathsf{T}$	9,7 dsj
	11.	$\neg(\mathsf{J}[C] = \mathsf{T} \triangledown \mathsf{J}[B] = \mathsf{T})$	10 dem
	12.	$\mathsf{J}[{\sim}A] \neq \mathsf{T}$	3 cnj
	13.	$\mathsf{J}[A] = \mathsf{T}$	12 ST($\sim$)
	14.	$\mathsf{J}[A \to (B \vee C)] = \mathsf{T}$	3 cnj
	15.	$\mathsf{J}[A] \neq \mathsf{T} \triangledown \mathsf{J}[B \vee C] = \mathsf{T}$	14 ST($\to$)
	16.	$\mathsf{J}[B \vee C] = \mathsf{T}$	13,15 dsj
	17.	$\mathsf{J}[B] = \mathsf{T} \triangledown \mathsf{J}[C] = \mathsf{T}$	16 ST$'$($\vee$)
	18.	$\mathsf{J}[C] = \mathsf{T} \triangledown \mathsf{J}[B] = \mathsf{T}$	17 com
	19.	$A \to (B \vee C), C \leftrightarrow B, {\sim}C \vDash_s {\sim}A$	1-18 neg

Suppose $A \to (B \vee C)$, $C \leftrightarrow B$, ${\sim}C \nvDash_s {\sim}A$; then by SV there is some I such that $\mathsf{I}[A \to (B \vee C)] = \mathsf{T}$, and $\mathsf{I}[C \leftrightarrow B] = \mathsf{T}$, and $\mathsf{I}[{\sim}C] = \mathsf{T}$, but $\mathsf{I}[{\sim}A] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[A \to (B \vee C)] = \mathsf{T}$, and $\mathsf{J}[C \leftrightarrow B] = \mathsf{T}$, and $\mathsf{J}[{\sim}C] = \mathsf{T}$, but $\mathsf{J}[{\sim}A] \neq \mathsf{T}$. Since $\mathsf{J}[{\sim}C] = \mathsf{T}$, by ST($\sim$), $\mathsf{J}[C] \neq \mathsf{T}$; so either $\mathsf{J}[C] \neq \mathsf{T}$ or $\mathsf{J}[B] \neq \mathsf{T}$; so it is not the case that both $\mathsf{J}[C] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$. But $\mathsf{J}[C \leftrightarrow B] = \mathsf{T}$; so by ST$'$($\leftrightarrow$), both $\mathsf{J}[C] = \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$, or both $\mathsf{J}[C] \neq \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$; but not the former, so $\mathsf{J}[C] \neq \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$; so it is not the case that either $\mathsf{J}[C] = \mathsf{T}$ or $\mathsf{J}[B] = \mathsf{T}$. $\mathsf{J}[{\sim}A] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[A] = \mathsf{T}$. But $\mathsf{J}[A \to (B \vee C)] = \mathsf{T}$; so by ST($\to$), $\mathsf{J}[A] \neq \mathsf{T}$ or $\mathsf{J}[B \vee C] = \mathsf{T}$; but $\mathsf{J}[A] = \mathsf{T}$; so $\mathsf{J}[B \vee C] = \mathsf{T}$; so by ST$'$($\vee$), $\mathsf{J}[B] = \mathsf{T}$ or $\mathsf{J}[C] = \mathsf{T}$; so either $\mathsf{J}[C] = \mathsf{T}$ or $\mathsf{J}[B] = \mathsf{T}$. But this is impossible; reject the assumption: $A \to (B \vee C)$, $C \leftrightarrow B$, ${\sim}C \nvDash_s {\sim}A$.

Though the formalized derivations are useful to discipline the way we reason, in the end, you may find the written versions to be both quicker, and easier to follow. As you work the exercises, try to free yourself from the formalized derivations to work the informal versions independently — though you should continue to use the formalized versions as a check for your work.

*E7.4. Complete the demonstration of derived clauses of ST$'$ by completing the demonstration for dst from right-to-left, and providing non-formalized reasonings for both the positive and negative parts of ST$'$($\vee$) and ST$'$($\leftrightarrow$).

E7.5. Using ST($\mid$) as above on p. 333, produce non-formalized reasonings to show each of the following. Again, you may or may not find formalized derivations helpful — but your reasoning should be no less clean than that guided by the rules. Hint, by ST($\mid$), $\mathsf{I}[\mathcal{P} \mid \mathcal{Q}] \neq \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$.

a. $\mathsf{I}[\mathcal{P} \mid \mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$

*b. $\mathsf{I}[\mathcal{P} \mid (\mathcal{Q} \mid \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$

c. $\mathsf{I}[(\mathcal{P} \mid \mathcal{P}) \mid (\mathcal{Q} \mid \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{T}$

d. $\mathsf{I}[(\mathcal{P} \mid \mathcal{Q}) \mid (\mathcal{P} \mid \mathcal{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P} \wedge \mathcal{Q}] = \mathsf{T}$

E7.6. Produce non-formalized reasoning to demonstrate each of the following.

a. $A \rightarrow (B \wedge C), C \leftrightarrow B, \sim C \vDash_s \sim A$

*b. $\sim(A \leftrightarrow B), \sim A, \sim B \vDash_s C \wedge \sim C$

*c. $\sim(\sim A \wedge \sim B) \nvDash_s A \wedge B$

d. $\sim\sim A \rightarrow \sim\sim B, \sim B \rightarrow \sim A \nvDash_s B \rightarrow A$

e. $A \wedge (B \rightarrow C) \nvDash_s (A \wedge C) \vee (A \wedge B)$

f. $[(C \vee D) \wedge B] \rightarrow A, D \vDash_s B \rightarrow A$

g. $\vDash_s [A \vee ((C \rightarrow \sim B) \wedge \sim A)] \vee \sim A$

h. $D \rightarrow (A \rightarrow B), \sim A \rightarrow \sim D, C \wedge D \vDash_s B$

i. $(\sim A \vee B) \rightarrow (C \wedge D), \sim(\sim A \vee B) \nvDash_s \sim(C \wedge D)$

j. $A \wedge (B \vee C), (\sim C \vee D) \wedge (D \rightarrow \sim D) \vDash_s A \wedge B$

7.3 Quantificational

So far, we might have obtained sentential results for validity and invalidity by truth tables. But our method positions us to make progress for the quantificational case, compared to what we were able to do before. Again, we will depend on, and gradually expand our formalized system as a guide.

7.3.1 Satisfaction

Given what we have done, it is easy to state definition SF for satisfaction as it applies to sentence letters, $\sim$, and $\to$. In this case, as described in chapter 4, we are reasoning about *satisfaction*, and satisfaction depends not just on interpretations, but on interpretations with variable assignments. For $\mathscr{S}$ an arbitrary sentence letter and $\mathscr{P}$ and $\mathscr{Q}$ any formulas, where $\mathsf{I_d}$ is an interpretation with variable assignment,

SF (s)	$\mathsf{I_d}[\mathscr{S}] = \mathsf{S} \Leftrightarrow \mathsf{I}[\mathscr{S}] = \mathsf{T}$	$\mathsf{I_d}[\mathscr{S}] \neq \mathsf{S} \Leftrightarrow \mathsf{I}[\mathscr{S}] \neq \mathsf{T}$
($\sim$)	$\mathsf{I_d}[\sim\mathscr{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] \neq \mathsf{S}$	$\mathsf{I_d}[\sim\mathscr{P}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] = \mathsf{S}$
($\to$)	$\mathsf{I_d}[\mathscr{P} \to \mathscr{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$	$\mathsf{I_d}[\mathscr{P} \to \mathscr{Q}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathscr{Q}] \neq \mathsf{S}$

Again, you should recognize this as a simple restatement of SF from p. 120. Rules for manipulating the definitions remain as before. Already, then, we can produce derived clauses for $\vee$, $\wedge$ and $\leftrightarrow$.

SF′ ($\vee$) $\mathsf{I_d}[(\mathscr{P} \vee \mathscr{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathscr{P} \vee \mathscr{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathscr{Q}] \neq \mathsf{S}$

($\wedge$) $\mathsf{I_d}[(\mathscr{P} \wedge \mathscr{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathscr{P} \wedge \mathscr{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] \neq \mathsf{S}$

($\leftrightarrow$) $\mathsf{I_d}[(\mathscr{P} \leftrightarrow \mathscr{Q})] = \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathscr{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}) \triangledown (\mathsf{I_d}[\mathscr{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathscr{Q}] \neq \mathsf{S})$

$\mathsf{I_d}[(\mathscr{P} \leftrightarrow \mathscr{Q})] \neq \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathscr{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathscr{Q}] \neq \mathsf{S}) \triangledown (\mathsf{I_d}[\mathscr{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathscr{Q}] = \mathsf{S})$

All these are are like ones from before. For the first,

(P)

1.	$\mathsf{I_d}[\mathscr{P} \vee \mathscr{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\sim\mathscr{P} \to \mathscr{Q}] = \mathsf{S}$	abv
2.	$\mathsf{I_d}[\sim\mathscr{P} \to \mathscr{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\sim\mathscr{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$	SF($\to$)
3.	$\mathsf{I_d}[\sim\mathscr{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$	SF($\sim$)
4.	$\mathsf{I_d}[\mathscr{P} \vee \mathscr{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathscr{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$	1,2,3 bcnd

Again, line (3) results from an implicit $\mathsf{I_d}[\sim\mathscr{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\sim\mathscr{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$ with ST($\sim$) as a replacement rule, substituting $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$ for $\mathsf{I_d}[\sim\mathscr{P}] \neq \mathsf{S}$ on the right-hand side. The informal reasoning is straightforward.

> By abv, $\mathsf{I_d}[\mathscr{P} \vee \mathscr{Q}] = \mathsf{S}$ iff $\mathsf{I_d}[\sim\mathscr{P} \to \mathscr{Q}] = \mathsf{S}$; by SF($\to$), iff $\mathsf{I_d}[\sim\mathscr{P}] \neq \mathsf{S}$ or $\mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$; by SF($\sim$), iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$ or $\mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$. So $\mathsf{I_d}[\mathscr{P} \vee \mathscr{Q}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$ or $\mathsf{I_d}[\mathscr{Q}] = \mathsf{S}$.

The reasoning is as before, except that our condition for satisfaction depends on an interpretation with variable assignment, rather than an interpretation alone.

Of course, given these definitions, we can use them in our reasoning. As a simple example, let us demonstrate that if $\mathsf{I_d}[\mathscr{P} \vee \mathscr{Q}] = \mathsf{S}$ and $\mathsf{I_d}[\sim\mathscr{Q}] = \mathsf{S}$, then $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$.

(Q)

1.	$\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S} \mathbin{\vartriangle} \mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$	assp
2.	$\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$	1 cnj
3.	$\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \mathbin{\triangledown} \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$	2 SF′(∨)
4.	$\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$	1 cnj
5.	$\mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$	4 SF(∼)
6.	$\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$	3,5 dsj
7.	$(\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S} \mathbin{\vartriangle} \mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}) \Rightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S}$	1-6 cnd

Suppose $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$ and $\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$. From the former, by SF′(∨), $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ or $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$; but $\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$; so by SF(∼), $\mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$; so $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$. So if $\mathsf{I_d}[\mathcal{P} \vee \mathcal{Q}] = \mathsf{S}$ and $\mathsf{I_d}[\sim\mathcal{Q}] = \mathsf{S}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$.

Again, basic reasoning is as in the sentential case, except that we carry along reference to variable assignments.

Observe that, given $\mathsf{I}[A] = \mathsf{T}$ for a sentence letter A, to show that $\mathsf{I_d}[A \vee B] = \mathsf{S}$, we reason,

(R)

1.	$\mathsf{I}[A] = \mathsf{T}$	ins
2.	$\mathsf{I_d}[A] = \mathsf{S}$	1 SF(s)
3.	$\mathsf{I_d}[A] = \mathsf{S} \mathbin{\triangledown} \mathsf{I_d}[B] = \mathsf{S}$	2 dsj
4.	$\mathsf{I_d}[A \vee B] = \mathsf{S}$	3 SF′(∨)

moving by SF(s) from the premise that the letter is true, to the result that it is satisfied, so that we are in a position to apply other clauses of the definition for satisfaction. SF applies to *satisfaction* not truth! So we have to bridge from one to the other before SF can apply!

This much should be straightforward, but let us pause to demonstrate derived clauses for satisfaction, and reinforce familiarity with the quantificational definition SF. As you work these and other problems, you may find the quantificational metalinguistic reference on p. 368 helpful.

E7.7. Produce formalized derivations and then informal reasoning to complete demonstrations for both positive and negative parts of derived clauses for SF′. Hint: you have been through the reasoning before!

*E7.8. Consider some $\mathsf{I_d}$ and suppose $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] \neq \mathsf{T}$ and $\mathsf{I}[C] = \mathsf{T}$. For each of the expressions in E7.1, produce the formalized and then informal reasoning to demonstrate either that it is or is not *satisfied* on $\mathsf{I_d}$.

7.3.2 Validity

In the quantificational case, there is a distinction between satisfaction and truth. We have been working with the definition for satisfaction. But validity is defined in terms of truth. So to reason about validity, we need a bridge from satisfaction to truth that applies beyond the case of sentence letters. For this, let 'A' be a metalinguistic universal quantifier. So, for example, $A\mathsf{d}(\mathsf{I_d}[\mathcal{P}] = \mathsf{S})$ says that any variable assignment d is such that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$. Then we have,

TI $\quad \mathsf{I}[\mathcal{P}] = \mathsf{T} \Leftrightarrow A\mathsf{d}(\mathsf{I_d}[\mathcal{P}] = \mathsf{S}) \qquad \mathsf{I}[\mathcal{P}] \neq \mathsf{T} \Leftrightarrow S\mathsf{d}(\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S})$

$\mathcal{P}$ is true on I iff it is satisfied for any variable assignment d. $\mathcal{P}$ is not true on I iff it is not satisfied for some variable assignment d. The definition QV is like SV.

QV $\quad \neg S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \vDash \mathcal{Q}$

$\quad S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \nvDash \mathcal{Q}$

An argument is quantificationally valid just in case there is no interpretation on which the premises are true and the conclusion is not. Of course, we are now talking about quantificational interpretations. Again, all of this repeats what was established in chapter 4.

To manipulate the universal quantifier, we will need some new rules. In chapter 6, we used ∀E to instantiate to *any* term — variable, constant, or otherwise. But ∀I was restricted — the idea being to generalize only on variables for truly *arbitrary* individuals. Corresponding restrictions are enforced here by the way terms are introduced. We generalize *from* variables for arbitrary individuals, but may instantiate *to* variables or constants of any kind. The universal rules are,

unv $\quad \dfrac{A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ of any type $\qquad \dfrac{\mathfrak{A}[\mathfrak{u}]}{A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary and new

If some $\mathfrak{A}$ is true for any $\mathfrak{t}$, then it is true for individual $\mathfrak{u}$. Thus we might move from the generalization, $A\mathsf{d}(\mathsf{I_d}[A] = \mathsf{S})$ to the particular claim $\mathsf{I_h}[A] = \mathsf{S}$ for assignment h. For the right-hand "introduction" rule, we require that $\mathfrak{u}$ be new in the sense required for ∀I in chapter 6. In particular, if $\mathfrak{u}$ is new to a derivation for goal $A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]$, $\mathfrak{u}$ will not appear free in any undischarged assumption when the universal rule is applied (typically, our derivations will be so simple that this will not be an issue). If we can show, say, $\mathsf{I_h}[A] = \mathsf{S}$ for arbitrary assignment h, then it is appropriate to move to the conclusion $A\mathsf{d}(\mathsf{I_d}[A] = \mathsf{S})$. We will also accept a metalinguistic quantifier negation, as in *ND+*.

qn $\quad \neg At\mathfrak{A} \Leftrightarrow St\neg\mathfrak{A} \qquad\qquad \neg St\mathfrak{A} \Leftrightarrow At\neg\mathfrak{A}$

With these definitions and rules, we are ready to reason about validity — at least for sentential forms. Suppose we want to show,

T7.1. $\mathcal{P}, \mathcal{P} \rightarrow \mathcal{Q} \vDash \mathcal{Q}$

1.	$\mathcal{P}, \mathcal{P} \rightarrow \mathcal{Q} \nvDash \mathcal{Q}$	assp
2.	$S\mathsf{I}(\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T})$	1 QV
3.	$\mathsf{J}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{J}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T} \vartriangle \mathsf{J}[\mathcal{Q}] \neq \mathsf{T})$	2 exs (J particular)
4.	$\mathsf{J}[Q] \neq \mathsf{T}$	3 cnj
5.	$S\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\mathcal{Q}] \neq \mathsf{S})$	4 TI
6.	$\mathsf{J}_{\mathsf{h}}[\mathcal{Q}] \neq \mathsf{S}$	5 exs (h particular)
7.	$\mathsf{J}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$	3 cnj
8.	$A\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S})$	7 TI
9.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S}$	8 unv
10.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{J}_{\mathsf{h}}[\mathcal{Q}] = \mathsf{S}$	9 SF($\rightarrow$)
11.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P}] \neq \mathsf{S}$	6,10 dsj
12.	$\mathsf{J}[\mathcal{P}] = \mathsf{T}$	3 cnj
13.	$A\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\mathcal{P}] = \mathsf{S})$	12 TI
14.	$\mathsf{J}_{\mathsf{h}}[\mathcal{P}] = \mathsf{S}$	13 unv
15.	$\mathcal{P}, \mathcal{P} \rightarrow \mathcal{Q} \vDash \mathcal{Q}$	1-14 neg

As usual, we begin with the assumption that the theorem is not valid, and apply the definition of validity for the result that the premises are true and the conclusion not. The goal is a contradiction. What is interesting are the applications of TI to bridge between truth and satisfaction. We begin by working on the conclusion. Since the conclusion is not true, by TI with exs we introduce a new variable assignment h on which the conclusion is not satisfied. Then, because the premises are true, by TI with unv the premises are satisfied on that very same assignment h. Then we use SF in the usual way. All this is like the strategy from *ND* by which we jump on existentials: If we had started with the premises, the requirement from exs that we instantiate to a *new* term would have forced a *different* variable assignment. But, by beginning with the conclusion, and coming with the universals from the premises after, we bring results into contact for contradiction.

> Suppose $\mathcal{P}$, $\mathcal{P} \rightarrow \mathcal{Q} \nvDash \mathcal{Q}$. Then by QV, there is some I such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$ but $\mathsf{I}[\mathcal{Q}] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$ but $\mathsf{J}[\mathcal{Q}] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J}_{\mathsf{d}}[\mathcal{Q}] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J}_{\mathsf{h}}[\mathcal{Q}] \neq \mathsf{S}$. But since $\mathsf{J}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$, by TI, for any d, $\mathsf{J}_{\mathsf{d}}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{S}$;

so by SF($\rightarrow$), $\mathsf{J_h}[\mathcal{P}] \neq \mathsf{S}$ or $\mathsf{J_h}[\mathcal{Q}] = \mathsf{S}$; so $\mathsf{J_h}[\mathcal{P}] \neq \mathsf{S}$. But since $\mathsf{J}[\mathcal{P}] = \mathsf{T}$, by TI, for any d, $\mathsf{J_d}[\mathcal{P}] = \mathsf{S}$; so $\mathsf{J_h}[\mathcal{P}] = \mathsf{S}$. This is impossible; reject the assumption: $\mathcal{P}$, $\mathcal{P} \rightarrow \mathcal{Q} \vDash \mathcal{Q}$.

Similarly we can show,

T7.2. $\vDash \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$

T7.3. $\vDash (\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})) \rightarrow ((\mathcal{O} \rightarrow \mathcal{P}) \rightarrow (\mathcal{O} \rightarrow \mathcal{Q}))$

T7.4. $\vDash (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow [(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}]$

T7.5. There is no interpretation I and formula $\mathcal{P}$ such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$.

Hint: Your goal is to show $\neg S\mathsf{I}(\mathsf{I}[\mathcal{P}] = \mathsf{T} \vartriangle \mathsf{I}[\sim\mathcal{P}] = \mathsf{T})$. You can get this by neg.

In each case, the pattern is the same: Bridge assumptions about truth to definition SF by TI with exs and unv. Reasoning with SF is as before. Given the requirement that the metalinguistic existential quantifier always be instantiated to a *new* variable or constant, it makes sense always to instantiate that which is not true, and so comes out as a metalinguistic existential, first, and then come with universals on "top" of terms already introduced. This is what we did above, and is like your derivation strategy in *ND*.

*E7.9. Produce formalized derivations and non-formalized reasoning to show that a,b,f,g,h from E7.6 are quantificationally valid.

E7.10. Provide demonstrations for T7.2, T7.3, T7.4 and T7.5 in the non-formalized style. Hint: You may or may not decide that formalized derivations would be helpful.

7.3.3 Terms and Atomics

So far, we have addressed only validity for sentential forms, and have not even seen the (r) and (∀) clauses for SF. We will get the quantifier clause in the next section. Here we come to the atomic clause for definition SF, but must first address the connection with interpretations *via* definition TA. For constant c, variable x, and complex term $h^n t_1 \dots t_n$, we say $\mathsf{I}[h^n]\langle \mathsf{a}_1 \dots \mathsf{a}_n\rangle$ is the thing the function $\mathsf{I}[h^n]$ associates with input $\langle \mathsf{a}_1 \dots \mathsf{a}_n\rangle$ (see p. 118).

TA (c) $\mathsf{I}_\mathsf{d}[c] = \mathsf{I}[c]$.

(v) $\mathsf{I}_\mathsf{d}[x] = \mathsf{d}[x]$.

(f) $\mathsf{I}_\mathsf{d}[h^n t_1 \dots t_n] = \mathsf{I}[h^n]\langle \mathsf{I}_\mathsf{d}[t_1] \dots \mathsf{I}_\mathsf{d}[t_n]\rangle$

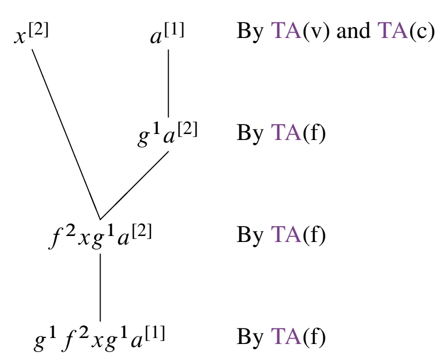
This is a direct restatement of the definition. To manipulate it, we need rules for equality.

eq $\mathfrak{t} = \mathfrak{t}$ $\qquad \mathfrak{t} = \mathfrak{u} \Leftrightarrow \mathfrak{u} = \mathfrak{t} \qquad \dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{u} = \mathfrak{v}}{\mathfrak{t} = \mathfrak{v}} \qquad \dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$

These should remind you of results from *ND*. We will allow generalized versions so that from $\mathfrak{t} = \mathfrak{u}$, $\mathfrak{u} = \mathfrak{v}$, and $\mathfrak{v} = \mathfrak{w}$, we might move directly to $\mathfrak{t} = \mathfrak{w}$. And we will not worry much about order around the equals sign so that, for example, we could move directly from $\mathfrak{t} = \mathfrak{u}$ and $\mathfrak{A}[\mathfrak{u}]$ to $\mathfrak{A}[\mathfrak{t}]$ without first converting $\mathfrak{t} = \mathfrak{u}$ to $\mathfrak{u} = \mathfrak{t}$ as required by the rule as stated. As in other cases, we will treat clauses from TA as both axioms and rules, though as usual, we typically take them as rules.

Let us consider first how this enables us to determine term assignments. Here is a relatively complex case. Suppose I has $\mathsf{U} = \{1, 2\}$, $\mathsf{I}[f^2] = \{\langle\langle 1, 1\rangle, 1\rangle, \langle\langle 1, 2\rangle, 1\rangle, \langle\langle 2, 1\rangle, 2\rangle, \langle\langle 2, 2\rangle, 2\rangle\}$, $\mathsf{I}[g^1] = \{\langle 1, 2\rangle, \langle 2, 1\rangle\}$, and $\mathsf{I}[a] = 1$. Recall that one-tuples are equated with their members so that $\mathsf{I}[g^1]$ is officially $\{\langle\langle 1\rangle, 2\rangle, \langle\langle 2\rangle, 1\rangle\}$. Suppose $\mathsf{d}[x] = 2$ and consider $\mathsf{I}_\mathsf{d}[g^1 f^2 x g^1 a]$. We might do this on a tree as in chapter 4.

(S)



Perhaps we whip through this on the tree. But the derivation follows the very same path, with explicit appeal to the definitions at every stage. In the derivation below, lines (1) - (4) cover the top row by application of TA(v) and TA(c). Lines (5) - (7) are like the second row, using the assignment to a with the interpretation of g^1 to determine the assignment to g^1a. Lines (8) - (10) cover the third row. And (11) - (13) use this to reach the final result.

1.	$\mathsf{I}[a] = 1$	ins (I particular)
2.	$\mathsf{I}_\mathsf{d}[a] = 1$	1 TA(c)
3.	$\mathsf{d}[x] = 2$	ins (d particular)
4.	$\mathsf{I}_\mathsf{d}[x] = 2$	3 TA(v)
5.	$\mathsf{I}_\mathsf{d}[g^1a] = \mathsf{I}[g^1]\langle 1\rangle$	2 TA(f)
6.	$\mathsf{I}[g^1]\langle 1\rangle = 2$	ins
7.	$\mathsf{I}_\mathsf{d}[g^1a] = 2$	5,6 eq
8.	$\mathsf{I}_\mathsf{d}[f^2xg^1a] = \mathsf{I}[f^2]\langle 2,2\rangle$	4,7 TA(f)
9.	$\mathsf{I}[f^2]\langle 2,2\rangle = 2$	ins
10.	$\mathsf{I}_\mathsf{d}[f^2xg^1a] = 2$	8,9 eq
11.	$\mathsf{I}_\mathsf{d}[g^1f^2xg^1a] = \mathsf{I}[g^1]\langle 2\rangle$	10 TA(f)
12.	$\mathsf{I}[g^1]\langle 2\rangle = 1$	ins
13.	$\mathsf{I}_\mathsf{d}[g^1f^2xg^1a] = 1$	11,12 eq

As with trees, to discover that to which a complex term is assigned, we find the assignment to the parts. Beginning with assignments to the parts, we work up to the assignment to the whole. Notice that assertions about the interpretation and the variable assignment are justified by ins. And notice the way we use TA as a rule at (2) and (4), and then again at (5), (8) and (11).

> $\mathsf{I}[a] = 1$; so by TA(c), $\mathsf{I}_\mathsf{d}[a] = 1$. And $\mathsf{d}[x] = 2$; so by TA(v), $\mathsf{I}_\mathsf{d}[x] = 2$. Since $\mathsf{I}_\mathsf{d}[a] = 1$, by TA(f), $\mathsf{I}_\mathsf{d}[g^1a] = \mathsf{I}[g^1]\langle 1\rangle$; but $\mathsf{I}[g^1]\langle 1\rangle = 2$; so $\mathsf{I}_\mathsf{d}[g^1a] = 2$. Since $\mathsf{I}_\mathsf{d}[x] = 2$ and $\mathsf{I}_\mathsf{d}[g^1a] = 2$, by TA(f), $\mathsf{I}_\mathsf{d}[f^2xg^1a] = \mathsf{I}[f^2]\langle 2,2\rangle$; but $\mathsf{I}[f^2]\langle 2,2\rangle = 2$; so $\mathsf{I}_\mathsf{d}[f^2xg^1a] = 2$. And from this, by TA(f), $\mathsf{I}_\mathsf{d}[g^1f^2xg^1a] = \mathsf{I}[g^1]\langle 2\rangle$; but $\mathsf{I}[g^1]\langle 2\rangle = 1$; so $\mathsf{I}_\mathsf{d}[g^1f^2xg^1a] = 1$.

With the ability to manipulate terms by TA, we can think about satisfaction and truth for arbitrary formulas without quantifiers. This brings us to SF(r). Say $\mathcal{R}^n$ is an $n-$place relation symbol, and $t_1 \dots t_n$ are terms.

SF(r) $\mathsf{I}_\mathsf{d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S} \Leftrightarrow \langle \mathsf{I}_\mathsf{d}[t_1] \dots \mathsf{I}_\mathsf{d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$

$\mathsf{I}_\mathsf{d}[\mathcal{R}^n t_1 \dots t_n] \neq \mathsf{S} \Leftrightarrow \langle \mathsf{I}_\mathsf{d}[t_1] \dots \mathsf{I}_\mathsf{d}[t_n]\rangle \notin \mathsf{I}[\mathcal{R}^n]$

This is a simple restatement of the definition from p. 120 in chapter 4. In fact, because of the simple negative version, we will apply the definition just in its positive form, and generate the negative case directly from it (as in NB from *ND+*).

Let us expand the above interpretation and variable assignment so that $\mathsf{I}[A^1] = \{2\}$ (or $\{\langle 2\rangle\}$) and $\mathsf{I}[B^2] = \{\langle 1,2\rangle, \langle 2,1\rangle\}$. Then $\mathsf{I}[Af^2xa] = \mathsf{S}$.

(T)

1.	$\mathsf{d}[x] = 2$	ins (d particular)
2.	$\mathsf{I_d}[x] = 2$	1 TA(v)
3.	$\mathsf{I}[a] = 1$	ins (I particular)
4.	$\mathsf{I_d}[a] = 1$	3 TA(c)
5.	$\mathsf{I_d}[f^2xa] = \mathsf{I}[f^2]\langle 2,1\rangle$	2,4 TA(f)
6.	$\mathsf{I}[f^2]\langle 2,1\rangle = 2$	ins
7.	$\mathsf{I_d}[f^2xa] = 2$	5,6 eq
8.	$\mathsf{I_d}[Af^2xa] = \mathsf{S} \Leftrightarrow \langle 2\rangle \in \mathsf{I}[A]$	7 SF(r)
9.	$\langle 2\rangle \in \mathsf{I}[A]$	ins
10.	$\mathsf{I_d}[Af^2xa] = \mathsf{S}$	8,9 bcnd

Again, this mirrors what we did with trees — moving through term assignments, to the value of the atomic. Observe that satisfaction is not the same as truth! Insofar as d is particular, (unv) does not apply for the result that Af^2xa is satisfied on every variable assignment, and so by TI that the formula is true. In this case, it is a simple matter to identify a variable assignment other than d on which the formula is not satisfied, and so to show that it is not true on I. Set $\mathsf{h}[x] = 1$.

1.	$\mathsf{h}[x] = 1$	ins (h particular)
2.	$\mathsf{I_h}[x] = 1$	1 TA(v)
3.	$\mathsf{I}[a] = 1$	ins (I particular)
4.	$\mathsf{I_h}[a] = 1$	3 TA(c)
5.	$\mathsf{I_h}[f^2xa] = \mathsf{I}[f^2]\langle 1,1\rangle$	2,4 TA(f)
6.	$\mathsf{I}[f^2]\langle 1,1\rangle = 1$	ins
7.	$\mathsf{I_h}[f^2xa] = 1$	5,6 eq
8.	$\mathsf{I_h}[Af^2xa] = \mathsf{S} \Leftrightarrow \langle 1\rangle \in \mathsf{I}[A]$	7 SF(r)
9.	$\langle 1\rangle \notin \mathsf{I}[A]$	ins
10.	$\mathsf{I_h}[Af^2xa] \neq \mathsf{S}$	8,9 bcnd
11.	$S\mathsf{d}(\mathsf{I_d}[Af^2xa] \neq \mathsf{S})$	10 exs
12.	$\mathsf{I}[Af^2xa] \neq \mathsf{T}$	11 TI

Given that it is not satisfied on the particular variable assignment h, (exs) and TI give the result that Af^2xa is not true. In this case, we simply *pick* the variable assignment we want: since the formula is not satisfied on this assignment, there is an assignment on which it is not satisfied; so it is not true. For a formula that is not a sentence, this is often the way to go. Just as it may be advantageous to find a particular interpretation to show invalidity, so it may be advantageous to seek out particular variable assignments for truth, in the case of open formulas.

> $\mathsf{h}[x] = 1$; so by TA(v), $\mathsf{I_h}[x] = 1$. And $\mathsf{I}[a] = 1$; so by TA(c), $\mathsf{I_h}[a] = 1$. So by TA(f), $\mathsf{I_h}[f^2xa] = \mathsf{I}[f^2]\langle 1,1\rangle$; but $\mathsf{I}[f^2]\langle 1,1\rangle = 1$; so $\mathsf{I_h}[f^2xa] = 1$. So by SF(r),

> $\mathsf{I_h}[Af^2xa] = \mathsf{S}$ iff $\langle 1\rangle \in \mathsf{I}[A]$; but $\langle 1\rangle \notin \mathsf{I}[A]$; so $\mathsf{I_h}[Af^2xa] \neq \mathsf{S}$. So there is a variable assignment d such that $\mathsf{I_d}[Af^2xa] \neq \mathsf{S}$; so by TI, $\mathsf{I}[Af^2xa] \neq \mathsf{T}$.

In contrast, even though it has free variables, Bxg^1x is true on this I. To show truth — a fact about *every* variable assignment — assume otherwise, and demonstrate a contradiction. This parallels our strategy for validity. Say o is a metalinguistic variable that ranges over members of U. In this case, it will be necessary to make an assertion by ins that $A\mathsf{o}(\mathsf{o} = 1 \triangledown \mathsf{o} = 2)$. This is clear enough, since $\mathsf{U} = \{1, 2\}$.

(U)

1.	$\mathsf{I}[Bxg^1x] \neq \mathsf{T}$	assp (I particular)
2.	$S\mathsf{d}(\mathsf{I_d}[Bxg^1x] \neq \mathsf{S})$	1 TI
3.	$\mathsf{I_h}[Bxg^1x] \neq \mathsf{S}$	2 exs (h particular)
4.	$A\mathsf{o}(\mathsf{o} = 1 \triangledown \mathsf{o} = 2)$	ins
5.	$\mathsf{I_h}[x] = 1 \triangledown \mathsf{I_h}[x] = 2$	4 unv
6.	$\quad \mathsf{I_h}[x] = 1$	assp
7.	$\quad \mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 1\rangle$	6 TA(f)
8.	$\quad \mathsf{I}[g^1]\langle 1\rangle = 2$	ins
9.	$\quad \mathsf{I_h}[g^1x] = 2$	7,8 eq
10.	$\quad \mathsf{I_h}[Bxg^1x] = \mathsf{S} \Leftrightarrow \langle 1, 2\rangle \in \mathsf{I}[B]$	6,9 SF(r)
11.	$\quad \langle 1, 2\rangle \notin \mathsf{I}[B]$	10,3 bcnd
12.	$\quad \langle 1, 2\rangle \in \mathsf{I}[B]$	ins
13.	$\mathsf{I_h}[x] \neq 1$	6-12 neg
14.	$\mathsf{I_h}[x] = 2$	5,13 dsj
15.	$\mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 2\rangle$	14 TA(f)
16.	$\mathsf{I}[g^1]\langle 2\rangle = 1$	ins
17.	$\mathsf{I_h}[g^1x] = 1$	15,16 eq
18.	$\mathsf{I_h}[Bxg^1x] = \mathsf{S} \Leftrightarrow \langle 2, 1\rangle \in \mathsf{I}[B]$	14,17 SF(r)
19.	$\langle 2, 1\rangle \notin \mathsf{I}[B]$	18,3 bcnd
20.	$\langle 2, 1\rangle \in \mathsf{I}[B]$	ins
21.	$\mathsf{I}[Bxg^1x] = \mathsf{T}$	1-20 neg

Up to this point, by ins we have made only particular claims about an assignment or interpretation, for example that $\langle 2, 1\rangle \in \mathsf{I}[B]$ or that $\mathsf{I}[g^1]\langle 2\rangle = 1$. This is the typical use of ins. In this case, however, at (4), we make a universal claim about U, any $\mathsf{o} \in \mathsf{U}$ is equal to 1 or 2. Since $\mathsf{I_h}[x]$ is a metalinguistic term, picking out some member of U, we instantiate the universal to it, with the result that $\mathsf{I_h}[x] = 1$ or $\mathsf{I_h}[x] = 2$. When U is small, this is often helpful: By ins we identify all the members of U; then we are in a position to argue about them individually. This argument works because we get the result no matter which thing $\mathsf{I_h}[x]$ happens to be.

> Suppose $\mathsf{I}[Bxg^1x] \neq \mathsf{T}$; then by TI, for some d, $\mathsf{I_d}[Bxg^1x] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{I_h}[Bxg^1x] \neq \mathsf{S}$. Since $\mathsf{U} = \{1, 2\}$, $\mathsf{I_h}[x] = 1$ or $\mathsf{I_h}[x] = 2$.

Suppose the former; then by TA(f), $\mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 1\rangle$; but $\mathsf{I}[g^1]\langle 1\rangle = 2$; so $\mathsf{I_h}[g^1x] = 2$; so by SF(r), $\mathsf{I_h}[Bxg^1x] = \mathsf{S}$ iff $\langle 1,2\rangle \in \mathsf{I}[B]$; so $\langle 1,2\rangle \notin \mathsf{I}[B]$; but $\langle 1,2\rangle \in \mathsf{I}[B]$; and this is impossible; reject the assumption; $\mathsf{I_h}[x] \neq 1$. So $\mathsf{I_h}[x] = 2$; so by TA(f), $\mathsf{I_h}[g^1x] = \mathsf{I}[g^1]\langle 2\rangle$; but $\mathsf{I}[g^1]\langle 2\rangle = 1$; so $\mathsf{I_h}[g^1x] = 1$; so by SF(r), $\mathsf{I_h}[Bxg^1x] = \mathsf{S}$ iff $\langle 2,1\rangle \in \mathsf{I}[B]$; so $\langle 2,1\rangle \notin \mathsf{I}[B]$. But $\langle 2,1\rangle \in \mathsf{I}[B]$. And this is impossible; reject the original assumption: $\mathsf{I}[Bxg^1x] = \mathsf{T}$.

To show that the formula is true, we assume otherwise. If there are no free variables, the argument may be straightforward. In this case with free variables, however, we are forced to reason individually about each of the possible assignments to x. It remains that we have been forced into cases. This is doable when U is small. We will have to consider other options when it is larger!

E7.11. Consider an I and d such that $\mathsf{U} = \{1,2\}$, $\mathsf{I}[a] = 1$, $\mathsf{I}[f^2] = \{\langle\langle 1,1\rangle,2\rangle, \langle\langle 1,2\rangle,1\rangle, \langle\langle 2,1\rangle,1\rangle, \langle\langle 2,2\rangle,2\rangle\}$, $\mathsf{I}[g^1] = \{\langle 1,1\rangle, \langle 2,1\rangle\}$, $\mathsf{d}[x] = 1$ and $\mathsf{d}[y] = 2$. Produce formalized derivations and non-formalized reasoning to determine the assignment $\mathsf{I_d}$ for each of the following.

a. a

b. g^1y

*c. g^1g^1x

d. f^2g^1ax

e. $f^2g^1af^2yx$

E7.12. Augment the above interpretation for E7.11 so that $\mathsf{I}[A^1] = \{1\}$ and $\mathsf{I}[B^2] = \{\langle 1,2\rangle, \langle 2,2\rangle\}$. Produce formalized derivations and non-formalized reasoning to demonstrate each of the following.

a. $\mathsf{I_d}[Ax] = \mathsf{S}$

*b. $\mathsf{I}[Byx] \neq \mathsf{T}$

c. $\mathsf{I}[Bg^1ay] \neq \mathsf{T}$

d. $\mathsf{I}[Aa] = \mathsf{T}$

e. $\mathsf{I}[\sim Bxg^1x] = \mathsf{T}$

7.3.4 Quantifiers

We are finally ready to think more generally about validity and truth for quantifier forms. For this, we will complete our formalized system by adding the quantifier clause to definition SF.

SF(∀) $\mathsf{I_d}[\forall x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathcal{A}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$ $\qquad$ $\mathsf{I_d}[\forall x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] \neq \mathsf{S})$

This is a simple statement of the definition from p. 120. Again, we treat the metalinguistic individual variable 'o' as implicitly restricted to the members of U (for any $\mathsf{o} \in \mathsf{U} \ldots$). You should think about this in relation to trees: From $\mathsf{I_d}[\forall x\mathcal{P}]$ there are branches with $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}]$ for each object $\mathsf{o} \in \mathsf{U}$. The universal is satisfied when each branch is satisfied; not satisfied when some branch is unsatisfied. That is what is happening above. We have the derived clause too.

SF′(∃) $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$ $\qquad$ $\mathsf{I_d}[\exists x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathcal{A}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] \neq \mathsf{S})$

The existential is satisfied when some branch is satisfied; not satisfied when every branch is not satisfied. For the positive form,

(V)

1.	$\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\sim\forall x\sim\mathcal{P}] = \mathsf{S}$	abv
2.	$\mathsf{I_d}[\sim\forall x\sim\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\forall x\sim\mathcal{P}] \neq \mathsf{S}$	SF(∼)
3.	$\mathsf{I_d}[\forall x\sim\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\sim\mathcal{P}] \neq \mathsf{S})$	SF(∀)
4.	$\mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\sim\mathcal{P}] \neq \mathsf{S}) \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$	SF(∼)
5.	$\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathcal{S}\mathsf{o}(\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S})$	1,2,3,4 bcnd

By abv, $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\sim\forall x\sim\mathcal{P}] = \mathsf{S}$; by SF(∼) iff $\mathsf{I_d}[\forall x\sim\mathcal{P}] \neq \mathsf{S}$; by SF(∀), iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\sim\mathcal{P}] \neq \mathsf{S}$; by SF(∼), iff for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S}$. So $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S}$ iff there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S}$.

Recall that we were not able to use trees to demonstrate validity in the quantificational case, because there were too many interpretations to have trees for all of them, and because universes may have too many members to have branches for all their members. But this is not a special difficulty for us now. For a simple case, let us show that $\vDash \forall x(Ax \to Ax)$.

$$
\text{(W)}\quad
\begin{array}{rl|l}
1. & \not\models \forall x(Ax \to Ax) & \text{assp} \\
2. & \mathcal{S}\mathsf{I}(\mathsf{I}[\forall x(Ax \to Ax)] \neq \mathsf{T}) & 1\ \text{QV} \\
3. & \mathsf{J}[\forall x(Ax \to Ax)] \neq \mathsf{T} & 2\ \text{exs (J particular)} \\
4. & \mathcal{S}\mathsf{d}(\mathsf{J_d}[\forall x(Ax \to Ax)] \neq \mathsf{S}) & 3\ \text{TI} \\
5. & \mathsf{J_h}[\forall x(Ax \to Ax)] \neq \mathsf{S} & 4\ \text{exs (h particular)} \\
6. & \mathcal{S}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax \to Ax] \neq \mathsf{S}) & 5\ \text{SF}(\forall) \\
7. & \mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax \to Ax] \neq \mathsf{S} & 6\ \text{exs (m particular)} \\
8. & \mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S} \mathbin{\vartriangle} \mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S} & 7\ \text{SF}(\to) \\
9. & \mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S} & 8\ \text{cnj} \\
10. & \mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S} & 8\ \text{cnj} \\
11. & \models \forall x(Ax \to Ax) & 1\text{-}10\ \text{neg}
\end{array}
$$

If $\forall x(Ax \to Ax)$ is not valid, there has to be *some* I on which it is not true. If $\forall x(Ax \to Ax)$ is not true on some I, there has to be some d on which it is not satisfied. And if the universal is not satisfied, there has to be some $\mathsf{o} \in \mathsf{U}$ for which the corresponding "branch" is not satisfied. But this is impossible — for we cannot have a branch where this is so.

> Suppose $\not\models \forall x(Ax \to Ax)$; then by QV, there is some I such that $\mathsf{I}[\forall x(Ax \to Ax)] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[\forall x(Ax \to Ax)] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{J_d}[\forall x(Ax \to Ax)] \neq \mathsf{S}$. Let h be a particular assignment of this sort; then $\mathsf{J_h}[\forall x(Ax \to Ax)] \neq \mathsf{S}$; so by SF($\forall$), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax \to Ax] \neq \mathsf{S}$. Let m be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax \to Ax] \neq \mathsf{S}$; so by SF($\to$), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S}$ and $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S}$. But this is impossible; reject the assumption: $\models \forall x(Ax \to Ax)$.

Notice, again, that the general strategy is to instantiate metalinguistic existential quantifiers as quickly as possible. Contradictions tend to arise at the level of atomic expressions and individuals.

Here is a case that is similar, but somewhat more involved. We show, $\forall x(Ax \to Bx), \exists x Ax \models \exists z Bz$. Here is a start.

(X)

1.	$\forall x(Ax \to Bx), \exists xAx \nvDash \exists zBz$	assp
2.	$\mathcal{S}\mathsf{I}(\mathsf{I}[\forall x(Ax \to Bx)] = \mathsf{T} \vartriangle \mathsf{I}[\exists xAx] = \mathsf{T} \vartriangle \mathsf{I}[\exists zBz] \neq \mathsf{T})$	1 QV
3.	$\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T} \vartriangle \mathsf{J}[\exists xAx] = \mathsf{T} \vartriangle \mathsf{J}[\exists zBz] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[\exists zBz] \neq \mathsf{T}$	3 cnj
5.	$\mathcal{S}\mathsf{d}(\mathsf{J_d}[\exists zBz] \neq \mathsf{S})$	4 TI
6.	$\mathsf{J_h}[\exists zBz] \neq \mathsf{S}$	5 exs (h particular)
7.	$\mathsf{J}[\exists xAx] = \mathsf{T}$	3 cnj
8.	$\mathcal{A}\mathsf{d}(\mathsf{J_d}[\exists xAx] = \mathsf{S})$	7 TI
9.	$\mathsf{J_h}[\exists xAx] = \mathsf{S}$	8 unv
10.	$\mathcal{S}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[Ax] = \mathsf{S})$	9 SF′(∃)
11.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Ax] = \mathsf{S}$	10 exs (m particular)
12.	$\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T}$	3 cnj
13.	$\mathcal{A}\mathsf{d}(\mathsf{J_d}[\forall x(Ax \to Bx)] = \mathsf{S})$	12 TI
14.	$\mathsf{J_h}[\forall x(Ax \to Bx)] = \mathsf{S}$	13 unv
15.	$\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[Ax \to Bx] = \mathsf{S})$	14 SF(∀)
16.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Ax \to Bx] = \mathsf{S}$	15 unv
17.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Ax] \neq \mathsf{S} \triangledown \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Bx] = \mathsf{S}$	16 SF(→)
18.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Bx] = \mathsf{S}$	17,11 dsj
19.	$\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(z\vert\mathsf{o})}[Bz] \neq \mathsf{S})$	6 SF′(∃)
20.	$\mathsf{J}_{\mathsf{h}(z\vert\mathsf{m})}[Bz] \neq \mathsf{S}$	19 unv

Note again the way we work with the metalinguistic quantifiers: We begin with the conclusion, because it is the one that requires a particular variable assignment; the premises can then be instantiated to that same assignment. Similarly, with that particular variable assignment on the table, we focus on the second premise, because it is the one that requires an instantiation to a particular individual. The other premise and the conclusion then come in later with universal quantifications that go onto the same thing. Also, $\mathsf{h}(x|\mathsf{m})[Ax] = \mathsf{S}$ contradicts $\mathsf{h}(x|\mathsf{m})[Ax] \neq \mathsf{S}$; this justifies dsj at (18). However $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$ at (18) does not contradict $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[Bz] \neq \mathsf{S}$ at (20). There would have been a contradiction if the variable had been the same. But it is not. However, with the distinct variables, we can bring out the contradiction by "forcing the result into the interpretation" as follows.

21.	$\mathsf{h}(x\vert\mathsf{m})[x] = \mathsf{m}$	ins
22.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = \mathsf{m}$	21 TA(v)
23.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Bx] = \mathsf{S} \Leftrightarrow \mathsf{m} \in \mathsf{J}[B]$	22 SF(r)
24.	$\mathsf{m} \in \mathsf{J}[B]$	23,18 bcnd
25.	$\mathsf{h}(z\vert\mathsf{m})[z] = \mathsf{m}$	ins
26.	$\mathsf{J}_{\mathsf{h}(z\vert\mathsf{m})}[z] = \mathsf{m}$	25 TA(v)
27.	$\mathsf{J}_{\mathsf{h}(z\vert\mathsf{m})}[Bz] = \mathsf{S} \Leftrightarrow \mathsf{m} \in \mathsf{J}[B]$	26 SF(r)
28.	$\mathsf{m} \notin \mathsf{J}[B]$	27,20 bcnd
29.	$\forall x(Ax \to Bx), \exists xAx \vDash \exists zBz$	1-28 neg

The assumption that the argument is not valid leads to the result that there is some interpretation J and $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{m} \in \mathsf{J}[B]$ and $\mathsf{m} \notin \mathsf{J}[B]$; so there can be no such interpretation, and the argument is quantificationally valid. Observe that, though we do not know anything else about h, simple inspection reveals that $\mathsf{h}(x|\mathsf{m})$ assigns object m to x. So we allow ourselves to assert it at (21) by ins; and similarly at (25). This pattern of moving from facts about satisfaction, to facts about the interpretation is typical.

> Suppose $\forall x(Ax \to Bx), \exists x Ax \nvDash \exists z Bz$; then by QV, there is some I such that $\mathsf{I}[\forall x(Ax \to Bx)] = \mathsf{T}$ and $\mathsf{I}[\exists x Ax] = \mathsf{T}$ but $\mathsf{I}[\exists z Bz] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T}$ and $\mathsf{J}[\exists x Ax] = \mathsf{T}$ but $\mathsf{J}[\exists z Bz] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J_d}[\exists z Bz] \neq \mathsf{S}$. Let h be a particular assignment of this sort; then $\mathsf{J_h}[\exists z Bz] \neq \mathsf{S}$. Since $\mathsf{J}[\exists x Ax] = \mathsf{T}$, by TI, for any d, $\mathsf{J_d}[\exists x Ax] = \mathsf{S}$; so $\mathsf{J_h}[\exists x Ax] = \mathsf{S}$; so by SF$'$($\exists$) there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax] = \mathsf{S}$. Let m be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] = \mathsf{S}$. Since $\mathsf{J}[\forall x(Ax \to Bx)] = \mathsf{T}$, by TI, for any d, $\mathsf{J_d}[\forall x(Ax \to Bx)] = \mathsf{S}$; so $\mathsf{J_h}[\forall x(Ax \to Bx)] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Ax \to Bx] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax \to Bx] = \mathsf{S}$; so by SF($\to$), either $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Ax] \neq \mathsf{S}$ or $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$; $\mathsf{h}(x|\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[x] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Bx] = \mathsf{S}$ iff $\mathsf{m} \in \mathsf{J}[B]$; so $\mathsf{m} \in \mathsf{J}[B]$. But since $\mathsf{J_d}[\exists z Bz] \neq \mathsf{S}$, by SF$'$($\exists$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(z|\mathsf{o})}[Bz] \neq \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[Bz] \neq \mathsf{S}$; $\mathsf{h}(z|\mathsf{m})[z] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[z] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(z|\mathsf{m})}[Bz] = \mathsf{S}$ iff $\mathsf{m} \in \mathsf{J}[B]$; so $\mathsf{m} \notin \mathsf{J}[B]$. This is impossible; reject the assumption: $\forall x(Ax \to Bx), \exists x Ax \vDash \exists z Bz$.

Observe again the repeated use of the pattern that moves from truth through TI with the quantifier rules to satisfaction, so that SF gets a grip, and the pattern that moves through satisfaction to the interpretation. These should be nearly automatic.

Here is an example that is particularly challenging in the way quantifier rules apply. We show, $\exists x \forall y Axy \vDash \forall y \exists x Axy$.

	1.	$\exists x\forall yAxy \nvDash \forall y\exists xAxy$	assp
	2.	$S\mathsf{I}(\mathsf{I}[\exists x\forall yAxy] = \mathsf{T} \vartriangle \mathsf{I}[\forall y\exists xAxy] \neq \mathsf{T})$	1 QV
	3.	$\mathsf{J}[\exists x\forall yAxy] = \mathsf{T} \vartriangle \mathsf{J}[\forall y\exists xAxy] \neq \mathsf{T}$	2 exs (J particular)
	4.	$\mathsf{J}[\forall y\exists xAxy] \neq \mathsf{T}$	3 cnj
	5.	$S\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\forall y\exists xAxy] \neq \mathsf{S})$	4 TI
	6.	$\mathsf{J}_{\mathsf{h}}[\forall y\exists xAxy] \neq \mathsf{S}$	5 exs (h particular)
	7.	$S\mathsf{o}(\mathsf{J}_{\mathsf{h}(y\mid\mathsf{o})}[\exists xAxy] \neq \mathsf{S})$	6 SF($\forall$)
	8.	$\mathsf{J}_{\mathsf{h}(y\mid\mathsf{m})}[\exists xAxy] \neq \mathsf{S}$	7 exs (m particular)
	9.	$\mathsf{J}[\exists x\forall yAxy] = \mathsf{T}$	3 cnj
(Y)	10.	$A\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\exists x\forall yAxy] = \mathsf{S})$	9 TI
	11.	$\mathsf{J}_{\mathsf{h}}[\exists x\forall yAxy] = \mathsf{S}$	10 exs
	12.	$S\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\mid\mathsf{o})}[\forall yAxy] = \mathsf{S})$	11 SF$'$($\exists$)
	13.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n})}[\forall yAxy] = \mathsf{S}$	12 exs (n particular)
	14.	$A\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{o})}[Axy] = \mathsf{S})$	13 SF($\forall$)
	15.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{m})}[Axy] = \mathsf{S}$	14 unv
	16.	$A\mathsf{o}(\mathsf{J}_{\mathsf{h}(y\mid\mathsf{m},x\mid\mathsf{o})}[Axy] \neq \mathsf{S})$	8 SF$'$($\exists$)
	17.	$\mathsf{J}_{\mathsf{h}(y\mid\mathsf{m},x\mid\mathsf{n})}[Axy] \neq \mathsf{S}$	16 unv
	18.	$\mathsf{h}(y\mid\mathsf{m}, x\mid\mathsf{n}) = \mathsf{h}(x\mid\mathsf{n}, y\mid\mathsf{m})$	ins
	19.	$\mathsf{J}_{\mathsf{h}(x\mid\mathsf{n},y\mid\mathsf{m})}[Axy] \neq \mathsf{S}$	17,18 eq
	20.	$\exists x\forall yAxy \vDash \forall y\exists xAxy$	1-19 neg

When multiple quantifiers come off, variable assignments are simply modified again — just as with trees. Observe again that we instantiate the metalinguistic existential quantifiers before universals. Also, the different existential quantifiers go to *different* individuals, to respect the requirement that individuals from exs be *new*. The key to this derivation is getting out *both* metalinguistic existentials for m and n *before* applying the corresponding universals — and what makes the derivation difficult is seeing that this needs to be done. Strictly, the variable assignment at (15) is the same as the one at (17), only the names are variants of one another. Thus we observe by ins that the assignments are the same, and apply eq for the contradiction. Another approach would have been to push for contradiction at the level of the interpretation. Thus, after (17) we might have continued,

18.	$\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})[x] = \mathsf{n}$	ins
19.	$\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})[y] = \mathsf{m}$	ins
20.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})}[x] = \mathsf{n}$	18 TA(v)
21.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})}[y] = \mathsf{m}$	19 TA(v)
22.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})}[Axy] = \mathsf{S} \Leftrightarrow \langle \mathsf{n}, \mathsf{m}\rangle \in \mathsf{I}[A]$	20,21 SF(r)
23.	$\langle \mathsf{n}, \mathsf{m}\rangle \in \mathsf{I}[A]$	22,15 bcnd
24.	$\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})[x] = \mathsf{n}$	ins
25.	$\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})[y] = \mathsf{m}$	ins
26.	$\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})}[x] = \mathsf{n}$	24 TA(v)
27.	$\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})}[y] = \mathsf{m}$	25 TA(v)
28.	$\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})}[Axy] = \mathsf{S} \Leftrightarrow \langle \mathsf{n}, \mathsf{m}\rangle \in \mathsf{I}[A]$	26,27 SF(r)
29.	$\langle \mathsf{n}, \mathsf{m}\rangle \notin \mathsf{I}[A]$	28,17 bcnd

This takes more steps, but follows a standard pattern. And you want to be particularly good at this pattern. We use facts about satisfaction to say that individuals assigned to terms are, or are not, in the interpretation of the relation symbol. Something along these lines would have been required if the conclusion had been, say, $\forall w \exists z Azw$. Based on this latter strategy, here is the non-formalized version.

> Suppose $\exists x \forall y Axy \nvDash \forall y \exists x Axy$; then by QV there is some I such that $\mathsf{I}[\exists x \forall y Axy] = \mathsf{T}$ and $\mathsf{I}[\forall y \exists x Axy] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\exists x \forall y Axy] = \mathsf{T}$ and $\mathsf{J}[\forall y \exists x Axy] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J}_{\mathsf{d}}[\forall y \exists x Axy] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J}_{\mathsf{h}}[\forall y \exists x Axy] \neq \mathsf{S}$; so by SF($\forall$), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{o})}[\exists x Axy] \neq \mathsf{S}$; let m be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m})}[\exists x Axy] \neq \mathsf{S}$. Since $\mathsf{J}[\exists x \forall y Axy] = \mathsf{T}$, by TI for any d, $\mathsf{J}_{\mathsf{d}}[\exists x \forall y Axy] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}}[\exists x \forall y Axy] = \mathsf{S}$; so by SF$'$($\exists$), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[\forall y Axy] = \mathsf{S}$; let n be a particular individual of this sort; then $\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n})}[\forall y Axy] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{o})}[Axy] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})}[Axy] = \mathsf{S}$. $\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})[x] = \mathsf{n}$ and $\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})[y] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})}[x] = \mathsf{n}$ and $\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})}[y] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(x\vert\mathsf{n}, y\vert\mathsf{m})}[Axy] = \mathsf{S}$ iff $\langle \mathsf{n}, \mathsf{m}\rangle \in \mathsf{I}[A]$; so $\langle \mathsf{n}, \mathsf{m}\rangle \in \mathsf{I}[A]$. Since $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m})}[\exists x Axy] \neq \mathsf{S}$, by SF$'$($\exists$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{o})}[Axy] \neq \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})}[Axy] \neq \mathsf{S}$. $\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})[x] = \mathsf{n}$ and $\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})[y] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})}[x] = \mathsf{n}$ and $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})}[y] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(y\vert\mathsf{m}, x\vert\mathsf{n})}[Axy] = \mathsf{S}$ iff $\langle \mathsf{n}, \mathsf{m}\rangle \in \mathsf{I}[A]$; so $\langle \mathsf{n}, \mathsf{m}\rangle \notin \mathsf{I}[A]$. This is impossible; reject the assumption: $\exists x \forall y Axy \vDash \forall y \exists x Axy$.

Try reading that to your roommate or parents! If you have followed to this stage, you have accomplished something significant. These are important results, given that we wondered in chapter 4 how this sort of thing could be done at all.

Here is a last trick that can sometimes be useful. Suppose we are trying to show $\forall x Px \vDash Pa$. We will come to a stage, where we want to use the premise to in-

stantiate a variable o to the thing that is $\mathsf{J_h}[a]$. So we might move directly from $\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Px] = \mathsf{S})$ to $\mathsf{J}_{\mathsf{h}(x|\mathsf{J_h}[a])}[Px] = \mathsf{S}$ by unv. But this is ugly, and hard to follow. An alternative is allow a rule (def) that defines m as a metalinguistic term for the *same* object as $\mathsf{J_h}[a]$. The result is as follows.

(Z)

1.	$\forall xPx \nvDash Pa$	assp
2.	$\mathcal{S}\mathsf{I}(\mathsf{I}[\forall xPx] = \mathsf{T} \vartriangle \mathsf{I}[Pa] \neq \mathsf{T})$	1 QV
3.	$\mathsf{J}[\forall xPx] = \mathsf{T} \vartriangle \mathsf{J}[Pa] \neq \mathsf{T}$	2 exs (J particular)
4.	$\mathsf{J}[Pa] \neq \mathsf{T}$	3 cnj
5.	$\mathcal{S}\mathsf{d}(\mathsf{J_d}[Pa] \neq \mathsf{S})$	4 TI
6.	$\mathsf{J_h}[Pa] \neq \mathsf{S}$	5 exs (h particular)
7.	$\mathsf{J_h}[a] = \mathsf{m}$	def (m particular)
8.	$\mathsf{J_h}[Pa] = \mathsf{S} \Leftrightarrow \mathsf{m} \in \mathsf{I}[P]$	7 SF(r)
9.	$\mathsf{m} \notin \mathsf{I}[P]$	6,8 bcnd
10.	$\mathsf{J}[\forall xPx] = \mathsf{T}$	2 cnj
11.	$\mathcal{A}\mathsf{d}(\mathsf{J_d}[\forall xPx] = \mathsf{S})$	10 TI
12.	$\mathsf{J_h}[\forall xPx] = \mathsf{S}$	11 unv
13.	$\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[Px] = \mathsf{S})$	12 SF($\forall$)
14.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Px] = \mathsf{S}$	13 unv
15.	$\mathsf{h}(x\vert\mathsf{m})[x] = \mathsf{m}$	ins
16.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = \mathsf{m}$	15 TA(v)
17.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[Px] = \mathsf{S} \Leftrightarrow \mathsf{m} \in \mathsf{I}[P]$	16 SF(r)
18.	$\mathsf{m} \in \mathsf{I}[P]$	17,14 bcnd
19.	$\forall xPx \vDash Pa$	1-18 neg

The result adds a couple lines, but is perhaps easier to follow. Though an interpretation is not specified, we can be sure that $\mathsf{J_h}[a]$ is some particular member of U; we simply let m designate that individual, and instantiate the universal to it.

> Suppose $\forall xPx \nvDash Pa$; then by QV, there is some I such that $\mathsf{I}[\forall xPx] = \mathsf{T}$ and $\mathsf{I}[Pa] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\forall xPx] = \mathsf{T}$ and $\mathsf{J}[Pa] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{J_d}[Pa] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J_h}[Pa] \neq \mathsf{S}$; where $\mathsf{m} = \mathsf{J_h}[a]$, by SF(r), $\mathsf{J_h}[Pa] = \mathsf{S}$ iff $\mathsf{m} \in \mathsf{I}[P]$; so $\mathsf{m} \notin \mathsf{I}[P]$. Since $\mathsf{J}[\forall xPx] = \mathsf{T}$, by TI, for any d, $\mathsf{J_d}[\forall xPx] = \mathsf{S}$; so $\mathsf{J_h}[\forall xPx] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Px] = \mathsf{S}$; so $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Px] = \mathsf{S}$; $\mathsf{h}(x|\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[x] = \mathsf{m}$; so by SF(r), $\mathsf{J}_{\mathsf{h}(x|\mathsf{m})}[Px] = \mathsf{S}$ iff $\mathsf{m} \in \mathsf{I}[P]$; so $\mathsf{m} \in \mathsf{I}[P]$. This is impossible; reject the assumption: $\forall xPx \nvDash Pa$.

Since we can instantiate $\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x|\mathsf{o})}[Px] = \mathsf{S})$ to any object, we can instantiate it to the one that happens to be $\mathsf{J_h}[a]$. The extra name streamlines the process. One can always do without the name. But there is no harm introducing it when it will help.

At this stage, we have the tools for a proof of the following theorem, that will be useful for later chapters.

T7.6. For any I and $\mathcal{P}$, $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\forall x\mathcal{P}] = \mathsf{T}$

Hint: If $\mathcal{P}$ is satisfied for the arbitrary assignment, you may conclude that it is satisfied on one like $\mathsf{h}(x|\mathsf{m})$. In the other direction, if you can instantiate o to any object, you can instantiate it to the thing that is $\mathsf{h}[x]$. But by ins, h with *this* assigned to x, just *is* h. So after substitution, you can end up with the very same assignment as the one with which you started.

This result is interesting insofar as it underlies principles like A4 and Gen in *AD* or $\forall$E and $\forall$I in *ND*. We further explore this link in following chapters.

E7.13. Produce formalized derivations and non-formalized reasoning to demonstrate each of the following.

a. $\vDash \forall x(Ax \to {\sim}{\sim}Ax)$

b. $\vDash {\sim}\exists x(Ax \wedge {\sim}Ax)$

*c. $Pa \vDash \exists xPx$

d. $\forall x(Ax \wedge Bx) \vDash \forall yBy$

e. $\forall yPy \vDash \forall xPf^1x$

f. $\exists yAy \vDash \exists x(Ax \vee Bx)$

g. ${\sim}\forall x(Ax \to Dx) \vDash \exists x(Ax \wedge {\sim}Dx)$

h. $\forall x(Ax \to Bx), \forall x(Bx \to Cx) \vDash \forall x(Ax \to Cx)$

i. $\forall x\forall yAxy \vDash \forall y\forall xAxy$

j. $\forall x\exists y(Ay \to Bx) \vDash \forall x(\forall yAy \to Bx)$

*E7.14. Provide a demonstrations for (a) the negative form of SF$'$($\exists$) and then (b) T7.6, both in the non-formalized style. Hint: You may or may not decide that formalized derivations would be helpful.

Metalinguistic Quick Reference (quantificational)

DEFINITIONS:

SF (s) $\mathsf{I_d}[\mathcal{S}] = \mathsf{S} \Leftrightarrow \mathsf{I}[\mathcal{S}] = \mathsf{T}$

(r) $\mathsf{I_d}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S} \Leftrightarrow \langle \mathsf{I_d}[t_1] \ldots \mathsf{I_d}[t_n] \rangle \in \mathsf{I}[\mathcal{R}^n]$

($\sim$) $\mathsf{I_d}[\sim\mathcal{P}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S}$

$\mathsf{I_d}[\sim\mathcal{P}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S}$

($\to$) $\mathsf{I_d}[\mathcal{P} \to \mathcal{Q}] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[\mathcal{P} \to \mathcal{Q}] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\forall$) $\mathsf{I_d}[\forall x\mathcal{P}] = \mathsf{S} \Leftrightarrow A\mathsf{o}(\mathsf{I_{d(x|o)}}[\mathcal{P}] = \mathsf{S})$

$\mathsf{I_d}[\forall x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow S\mathsf{o}(\mathsf{I_{d(x|o)}}[\mathcal{P}] \neq \mathsf{S})$

SF′ ($\vee$) $\mathsf{I_d}[(\mathcal{P} \vee \mathcal{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathcal{P} \vee \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\wedge$) $\mathsf{I_d}[(\mathcal{P} \wedge \mathcal{Q})] = \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$

$\mathsf{I_d}[(\mathcal{P} \wedge \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow \mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \triangledown \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}$

($\leftrightarrow$) $\mathsf{I_d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] = \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S}) \triangledown (\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S})$

$\mathsf{I_d}[(\mathcal{P} \leftrightarrow \mathcal{Q})] \neq \mathsf{S} \Leftrightarrow (\mathsf{I_d}[\mathcal{P}] = \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] \neq \mathsf{S}) \triangledown (\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S} \vartriangle \mathsf{I_d}[\mathcal{Q}] = \mathsf{S})$

($\exists$) $\mathsf{I_d}[\exists x\mathcal{P}] = \mathsf{S} \Leftrightarrow S\mathsf{o}(\mathsf{I_{d(x|o)}}[\mathcal{P}] = \mathsf{S})$

$\mathsf{I_d}[\exists x\mathcal{P}] \neq \mathsf{S} \Leftrightarrow A\mathsf{o}(\mathsf{I_{d(x|o)}}[\mathcal{P}] \neq \mathsf{S})$

TA (c) $\mathsf{I_d}[c] = \mathsf{I}[c]$

(v) $\mathsf{I_d}[x] = \mathsf{d}[x]$

(f) $\mathsf{I_d}[h^n t_1 \ldots t_n] = \mathsf{I}[h^n]\langle \mathsf{I_d}[t_1] \ldots \mathsf{I_d}[t_n] \rangle$

TI $\mathsf{I}[\mathcal{P}] = \mathsf{T} \Leftrightarrow A\mathsf{d}(\mathsf{I_d}[\mathcal{P}] = \mathsf{S})$

$\mathsf{I}[\mathcal{P}] \neq \mathsf{T} \Leftrightarrow S\mathsf{d}(\mathsf{I_d}[\mathcal{P}] \neq \mathsf{S})$

QV $\neg S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \vDash \mathcal{Q}$

$S\mathsf{I}(\mathsf{I}[\mathcal{P}_1] = \mathsf{T} \vartriangle \ldots \vartriangle \mathsf{I}[\mathcal{P}_n] = \mathsf{T} \vartriangle \mathsf{I}[\mathcal{Q}] \neq \mathsf{T}) \Leftrightarrow \mathcal{P}_1 \ldots \mathcal{P}_n \nvDash \mathcal{Q}$

RULES:

All the rules from the sentential metalinguistic reference (p. 346) plus:

unv $\dfrac{A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$ $\mathfrak{u}$ of any type $\qquad \dfrac{\mathfrak{A}[\mathfrak{u}]}{A\mathfrak{t}\mathfrak{A}[\mathfrak{t}]}$ $\mathfrak{u}$ arbitrary and new

qn $\neg A\mathfrak{t}\mathfrak{A} \Leftrightarrow S\mathfrak{t}\neg\mathfrak{A}$ $\qquad \neg S\mathfrak{t}\mathfrak{A} \Leftrightarrow A\mathfrak{t}\neg\mathfrak{A}$

eq $\mathfrak{t} = \mathfrak{t}$ $\qquad \mathfrak{t} = \mathfrak{u} \Leftrightarrow \mathfrak{u} = \mathfrak{t}$ $\qquad \dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{u} = \mathfrak{v}}{\mathfrak{t} = \mathfrak{v}}$ $\qquad \dfrac{\mathfrak{t} = \mathfrak{u},\ \mathfrak{A}[\mathfrak{t}]}{\mathfrak{A}[\mathfrak{u}]}$

def Defines one metalinguistic term $\mathfrak{t}$ by another $\mathfrak{u}$ so that $\mathfrak{t} = \mathfrak{u}$.

7.3.5 Invalidity

We already have in hand concepts required for showing invalidity. Difficulties are mainly strategic and practical. As usual, for invalidity, the idea is to produce an interpretation, and show that it makes the premises true and the conclusion not. Here is a case parallel to one you worked with trees in homework from E4.14. We show $\forall x P f^1 x \not\vDash \forall x P x$. For the interpretation J set, $\mathsf{U} = \{1, 2\}$, $\mathsf{J}[P] = \{1\}$, $\mathsf{J}[f^1] = \{\langle 1, 1\rangle, \langle 2, 1\rangle\}$. We want to take advantage of the particular features of this interpretation to show that it makes the premise true and the conclusion not. Begin as follows.

(AA)

1.	$\mathsf{J}[\forall x P x] = \mathsf{T}$	assp (J particular)
2.	$\mathcal{A}\mathsf{d}(\mathsf{J_d}[\forall x P x] = \mathsf{S})$	1 TI
3.	$\mathsf{J_h}[\forall x P x] = \mathsf{S}$	2 unv (h particular)
4.	$\mathcal{A}\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\|\mathsf{o})}[P x] = \mathsf{S})$	3 SF($\forall$)
5.	$\mathsf{J}_{\mathsf{h}(x\|2)}[P x] = \mathsf{S}$	4 unv
6.	$\mathsf{h}(x\|2)[x] = 2$	ins
7.	$\mathsf{J}_{\mathsf{h}(x\|2)}[x] = 2$	6 TA(v)
8.	$\mathsf{J}_{\mathsf{h}(x\|2)}[P x] = \mathsf{S} \Leftrightarrow 2 \in \mathsf{J}[P]$	7 SF(r)
9.	$2 \in \mathsf{J}[P]$	8,5 bcnd
10.	$2 \notin \mathsf{J}[P]$	ins
11.	$\mathsf{J}[\forall x P x] \neq \mathsf{T}$	1-10 neg

This much is straightforward. We instantiate the metalinguistic universal quantifier to 2, because that is the individual which exposes the conclusion as not true. Now one option is to reason individually about each member of U. This is always possible, and sometimes necessary. Thus the argument is straightforward but tedious by methods we have seen before.

12.	$\mathsf{J}[\forall x P f^1 x] \neq \mathsf{T}$	assp
13.	$S\mathsf{d}(\mathsf{J}_{\mathsf{d}}[\forall x P f^1 x] \neq \mathsf{S})$	12 TI
14.	$\mathsf{J}_{\mathsf{h}}[\forall x P f^1 x] \neq \mathsf{S}$	13 exs (h particular)
15.	$S\mathsf{o}(\mathsf{J}_{\mathsf{h}(x\vert\mathsf{o})}[P f^1 x] \neq \mathsf{S})$	14 SF(∀)
16.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[P f^1 x] \neq \mathsf{S}$	15 exs (m particular)
17.	$\mathsf{h}(x\vert\mathsf{m})[x] = \mathsf{m}$	ins
18.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = \mathsf{m}$	17 TA(v)
19.	$A\mathsf{o}(\mathsf{o} = 1 \triangledown \mathsf{o} = 2)$	ins
20.	$\mathsf{m} = 1 \triangledown \mathsf{m} = 2$	19 unv
21.	$\quad \mathsf{m} = 1$	assp
22.	$\quad \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = 1$	18,21 eq
23.	$\quad \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1 x] = \mathsf{J}[f^1]\langle 1\rangle$	22 TA(f)
24.	$\quad \mathsf{J}[f^1]\langle 1\rangle = 1$	ins
25.	$\quad \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1 x] = 1$	24,23 eq
26.	$\quad \mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[P f^1 x] = \mathsf{S} \Leftrightarrow 1 \in \mathsf{J}[P]$	25 SF(r)
27.	$\quad 1 \notin \mathsf{J}[P]$	26,16 bcnd
28.	$\quad 1 \in \mathsf{J}[P]$	ins
29.	$\mathsf{m} \neq 1$	21-28 neg
30.	$\mathsf{m} = 2$	20,29 dsj
31.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[x] = 2$	18,30 eq
32.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1 x] = \mathsf{J}[f^1]\langle 2\rangle$	31 TA(f)
33.	$\mathsf{J}[f^1]\langle 2\rangle = 1$	ins
34.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1 x] = 1$	33,32 eq
35.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[P f^1 x] = \mathsf{S} \Leftrightarrow 1 \in \mathsf{J}[P]$	34 SF(r)
36.	$1 \notin \mathsf{J}[P]$	35,16 bcnd
37.	$1 \in \mathsf{J}[P]$	ins
38.	$\mathsf{J}[\forall x P f^1 x] = \mathsf{T}$	12-37 neg
39.	$\mathsf{J}[\forall x P f^1 x] = \mathsf{T} \vartriangle \mathsf{J}[\forall x P x] \neq \mathsf{T}$	11,38 cnj
40.	$S\mathsf{I}(\mathsf{I}[\forall x P f^1 x] = \mathsf{T} \vartriangle \mathsf{I}[\forall x P x] \neq \mathsf{T})$	39 exs
41.	$\forall x P f^1 x \not\vDash \forall x P x$	40 QV

m has to be some member of U, so we instantiate the universal at (19) to it, and reason about the cases individually. This reflects what we have done before.

But this interpretation is designed so that no matter what o may be, $\mathsf{I}[f^1]\langle\mathsf{o}\rangle = 1$. And, rather than the simple generalization about the universe of discourse, we might have generalized by ins about the interpretation of the function symbol itself. Thus, we might have substituted for lines (19) - (34) as follows.

19.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1 x] = \mathsf{J}[f^1]\langle\mathsf{m}\rangle$	18 TA(f)
20.	$A\mathsf{o}(\mathsf{J}[f^1]\langle\mathsf{o}\rangle) = 1$	ins
21.	$\mathsf{J}[f^1]\langle\mathsf{m}\rangle = 1$	20 unv
22.	$\mathsf{J}_{\mathsf{h}(x\vert\mathsf{m})}[f^1 x] = 1$	19,21 eq

picking up with (35) after. This is better! Before, we found the contradiction when m

was 1 and again when m was 2. But, in either case, the reason for the contradiction is that the function has output 1. So this version avoids the cases, by reasoning directly about the result from the function. Here is the non-formalized version on this latter strategy.

> Suppose $\mathsf{J}[\forall x Px] = \mathsf{T}$; then by TI, for any d, $\mathsf{J_d}[\forall x Px] = \mathsf{S}$; let h be a particular assignment; then $\mathsf{J_h}[\forall x Px] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J_{h(\mathit{x}|o)}}[Px] = \mathsf{S}$; so $\mathsf{J_{h(\mathit{x}|2)}}[Px] = \mathsf{S}$; $\mathsf{h}(x|2)[x] = 2$; so by TA(v), $\mathsf{J_{h(\mathit{x}|2)}}[x] = 2$; so by SF(r), $\mathsf{J_{h(\mathit{x}|2)}}[Px] = \mathsf{S}$ iff $2 \in \mathsf{J}[P]$; so $2 \in \mathsf{J}[P]$. But $2 \notin \mathsf{J}[P]$. This is impossible; reject the assumption: $\mathsf{J}[\forall x Px] \neq \mathsf{T}$.
>
> Suppose $\mathsf{J}[\forall x Pf^1x] \neq \mathsf{T}$; then by TI, for some d, $\mathsf{J_d}[\forall x Pf^1x] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{J_h}[\forall x Pf^1x] \neq \mathsf{S}$; so by SF($\forall$), for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{J_{h(\mathit{x}|o)}}[Pf^1x] \neq \mathsf{S}$; let m be a particular individual of this sort; then $\mathsf{J_{h(\mathit{x}|m)}}[Pf^1x] \neq \mathsf{S}$. $\mathsf{h}(x|\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{J_{h(\mathit{x}|m)}}[x] = \mathsf{m}$; so by TA(f), $\mathsf{J_{h(\mathit{x}|m)}}[f^1x] = \mathsf{J}[f^1]\langle \mathsf{m} \rangle$; but for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{J}[f^1]\langle \mathsf{o} \rangle = 1$; so $\mathsf{J}[f^1]\langle \mathsf{m} \rangle = 1$; so $\mathsf{J_{h(\mathit{x}|m)}}[f^1x] = 1$; so by SF(r), $\mathsf{J_{h(\mathit{x}|m)}}[Pf^1x] = \mathsf{S}$ iff $1 \in \mathsf{J}[P]$; so $1 \notin \mathsf{J}[P]$; but $1 \in \mathsf{J}[P]$. This is impossible; reject the assumption: $\mathsf{J}[\forall x Pf^1x] = \mathsf{T}$.
>
> So there is an interpretation I such that $\mathsf{I}[\forall x Pf^1x] = \mathsf{T}$ and $\mathsf{I}[\forall x Px] \neq \mathsf{T}$; so by QV, $\forall x Pf^1x \not\models \forall x Px$.

Reasoning about cases is possible, and sometimes necessary, when the universe is small. But it is often convenient to organize your reasoning by generalizations about the interpretation as above. Such generalizations are required when the universe is large.

Here is a case that requires such generalizations, insofar as the universe U for the interpretation to show invalidity has infinitely many members. We show $\forall x \forall y(Sx = Sy \rightarrow x = y) \not\models \exists x(Sx = \emptyset)$. First note that no interpretation with finite U makes the premise true and conclusion false. For suppose U has finitely many members and the successor function is represented by arrows as follows,

$$\mathsf{o_0} \longrightarrow \mathsf{o_1} \longrightarrow \mathsf{o_2} \longrightarrow \mathsf{o_3} \longrightarrow \mathsf{o_4} \longrightarrow \mathsf{o_5} \ldots \mathsf{o_n}$$

with $\mathsf{I}[\emptyset] = \mathsf{o_0}$. So $\mathsf{I}[S]$ includes $\langle \mathsf{o_0}, \mathsf{o_1} \rangle$, $\langle \mathsf{o_1}, \mathsf{o_2} \rangle$, $\langle \mathsf{o_2}, \mathsf{o_3} \rangle$, and so forth. What is paired with $\mathsf{o_n}$? It cannot be any of $\mathsf{o_1}$ through $\mathsf{o_n}$, or the premise is violated, because some one thing is the successor of different elements (you should see how this works). And if the conclusion is false, it cannot be $\mathsf{o_0}$ either. And similarly for any finite universe. But, as should be obvious by consideration of a standard interpretation of the symbols, the argument is not valid. To show this, let the interpretation be N, where,

> $\mathsf{U} = \{0, 1, 2 \ldots\}$

$\mathsf{N}[\emptyset] = 0$

$\mathsf{N}[S] = \{\langle 0,1\rangle, \langle 1,2\rangle, \langle 2,3\rangle \ldots\}$

$\mathsf{N}[=] = \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle \ldots\}$

First we show that $\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$. Note that we might have specified the interpretation for equality by saying something like, $\mathcal{A}\mathsf{o}\mathcal{A}\mathsf{p}(\langle \mathsf{o},\mathsf{p}\rangle \in \mathsf{N}[=] \Leftrightarrow \mathsf{o} = \mathsf{p})$. Similarly, the interpretation of S is such that no o has a successor equal to zero — $\mathcal{A}\mathsf{o}(\mathsf{N}[S]\langle \mathsf{o}\rangle \neq 0)$. We will simply appeal to these facts by ins in the following.

(AB)

1.	$\mathsf{N}[\exists x(Sx = \emptyset)] = \mathsf{T}$	assp (N particular)
2.	$\mathcal{A}\mathsf{d}(\mathsf{N_d}[\exists x(Sx = \emptyset)] = \mathsf{S})$	1 TI
3.	$\mathsf{N_h}[\exists x(Sx = \emptyset)] = \mathsf{S}$	2 exs (h particular)
4.	$\mathcal{S}\mathsf{o}(\mathsf{N}_{\mathsf{h}(x\mid\mathsf{o})}[Sx = \emptyset] = \mathsf{S})$	3 SF′(∃)
5.	$\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx = \emptyset] = \mathsf{S}$	4 exs (m particular)
6.	$\mathsf{N}[\emptyset] = 0$	ins
7.	$\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[\emptyset] = 0$	6 TA(c)
8.	$\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx = \emptyset] = \mathsf{S} \Leftrightarrow \langle \mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx], 0\rangle \in \mathsf{N}[=]$	7 SF(r)
9.	$\langle \mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx], 0\rangle \in \mathsf{N}[=]$	8,5 bcnd
10.	$\mathcal{A}\mathsf{o}\mathcal{A}\mathsf{p}(\langle \mathsf{o},\mathsf{p}\rangle \in \mathsf{N}[=] \Rightarrow \mathsf{o} = \mathsf{p})$	ins
11.	$\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx] = 0$	10,9 unv
12.	$\mathsf{h}(x\mid\mathsf{m})[x] = \mathsf{m}$	ins
13.	$\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[x] = \mathsf{m}$	12 TA(v)
14.	$\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx] = \mathsf{N}[S]\langle \mathsf{m}\rangle$	13 TA(f)
15.	$\mathsf{N}[S]\langle \mathsf{m}\rangle = 0$	11,14 eq
16.	$\mathcal{A}\mathsf{o}(\mathsf{N}[S]\langle \mathsf{o}\rangle \neq 0)$	ins
17.	$\mathsf{N}[S]\langle \mathsf{m}\rangle \neq 0$	16 unv
18.	$\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$	1-17 neg

Most of this is as usual. What is interesting is that at (10) we assert that for any o and p in U, if $\langle \mathsf{o},\mathsf{p}\rangle \in \mathsf{U}$, then $\mathsf{o} = \mathsf{p}$ by ins. This should be obvious from the initial (automatic) specification of $\mathsf{N}[=]$. And at (16) we assert that no o is such that $\langle \mathsf{o}, 0\rangle \in \mathsf{N}[S]$. Again, this should be clear from the specification of $\mathsf{N}[S]$. In this case, there is no way to instantiate the metalinguistic quantifiers to *every* member of U, on the pattern of what we have been able to do with two-member universes! But we do not have to, as the general facts are sufficient for the result.

Suppose $\mathsf{N}[\exists x(Sx = \emptyset)] = \mathsf{T}$; then by TI, for any d, $\mathsf{N_d}[\exists x(Sx = \emptyset)] = \mathsf{S}$; let h be some particular d; then $\mathsf{N_h}[\exists x(Sx = \emptyset)] = \mathsf{S}$; so by SF′(∃), for some $\mathsf{o} \in \mathsf{U}$, $\mathsf{N}_{\mathsf{h}(x\mid\mathsf{o})}[Sx = \emptyset] = \mathsf{S}$; let m be a particular individual of this sort; then $\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx = \emptyset] = \mathsf{S}$. $\mathsf{N}[\emptyset] = 0$; so by TA(c), $\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[\emptyset] = 0$; so by SF(r), $\mathsf{N}_{\mathsf{h}(x\mid\mathsf{m})}[Sx = \emptyset] = \mathsf{S}$

iff $\langle \mathsf{N}_{\mathsf{h}(x|\mathsf{m})}[Sx], 0\rangle \in \mathsf{N}[=]$; so $\langle \mathsf{N}_{\mathsf{h}(x|\mathsf{m})}[Sx], 0\rangle \in \mathsf{N}[=]$; but for any $\mathsf{o}, \mathsf{p} \in \mathsf{U}$, if $\langle \mathsf{o}, \mathsf{p}\rangle \in \mathsf{N}[=]$ then $\mathsf{o} = \mathsf{p}$; so $\mathsf{N}_{\mathsf{h}(x|\mathsf{m})}[Sx] = 0$. $\mathsf{h}(x|\mathsf{m})[x] = \mathsf{m}$; so by TA(v), $\mathsf{N}_{\mathsf{h}(x|\mathsf{m})}[x] = \mathsf{m}$; so by TA(f), $\mathsf{N}_{\mathsf{h}(x|\mathsf{m})}[Sx] = \mathsf{N}[S]\langle \mathsf{m}\rangle$; so $\mathsf{N}[S]\langle \mathsf{m}\rangle = 0$. But for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{N}[S]\langle \mathsf{o}\rangle \neq 0$; so $\mathsf{N}[S]\langle \mathsf{m}\rangle \neq 0$. This is impossible; reject the assumption: $\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$.

Given what we have already seen, this should be straightforward. Demonstration that $\mathsf{N}[\forall x \forall y(Sx = Sy \to x = y)] = \mathsf{T}$, and so that the argument is not valid, is left as an exercise. Hint: In addition to facts about equality, you may find it helpful to assert $A\mathsf{o}A\mathsf{p}(\mathsf{o} \neq \mathsf{p} \Rightarrow \mathsf{N}[S]\langle \mathsf{o}\rangle \neq \mathsf{N}[S]\langle \mathsf{p}\rangle)$. Be sure that you understand this, before you assert it! Of course, we have here something that could never have been accomplished with trees, insofar as the universe is infinite!

Recall that the interpretation of equality is the same across all interpretations. Thus our general assertion is possible in case of the arbitrary interpretation, and we are positioned to prove some last theorems.

T7.7. $\vDash (t = t)$

Hint: By ins for any I, and any $\mathsf{o} \in \mathsf{U}$, $\langle \mathsf{o}, \mathsf{o}\rangle \in \mathsf{N}[=]$. Given this, the argument is easy.

*T7.8. $\vDash (x_i = y) \to (h^n x_1 \dots x_i \dots x_n = h^n x_1 \dots y \dots x_n)$

Hint: If you have trouble with this, try showing a simplified version: $\vDash (x = y) \to (h^1 x = h^1 y)$.

T7.9. $\vDash (x_i = y) \to (\mathcal{R}^n x_1 \dots x_i \dots x_n \to \mathcal{R}^n x_1 \dots y \dots x_n)$

Hint: If you have trouble with this, try showing a simplified version: $\vDash (x = y) \to (Rx \to Ry)$.

At this stage, we have introduced a method for reasoning about semantic definitions. As you continue to work with the definitions, it should become increasingly clear how they fit together into a coherent (and pleasing) whole. In later chapters, we will leave the formalized system behind as we encounter further definitions in diverse contexts. But from this chapter you should have gained a solid grounding in the *sort* of thing we will want to do.

E7.15. Produce interpretations (with, if necessary, variable assignments) and then formalized derivations and non-formalized reasoning to show each of the following.

Theorems of Chapter 7

T7.1s $\mathcal{P}, \mathcal{P} \rightarrow \mathcal{Q} \vDash_s \mathcal{Q}$

T7.2s $\vDash_s \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$

T7.3s $\vDash_s (\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})) \rightarrow ((\mathcal{O} \rightarrow \mathcal{P}) \rightarrow (\mathcal{O} \rightarrow \mathcal{Q}))$

T7.4s $\vDash_s (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow [(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}]$

T7.1 $\mathcal{P}, \mathcal{P} \rightarrow \mathcal{Q} \vDash \mathcal{Q}$

T7.2 $\vDash \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$

T7.3 $\vDash (\mathcal{O} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})) \rightarrow ((\mathcal{O} \rightarrow \mathcal{P}) \rightarrow (\mathcal{O} \rightarrow \mathcal{Q}))$

T7.4 $\vDash (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow [(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}]$

T7.5 There is no interpretation I and formula $\mathcal{P}$ such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$.

T7.6 For any I and $\mathcal{P}$, $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\forall x \mathcal{P}] = \mathsf{T}$

T7.7 $\vDash (t = t)$

T7.8 $\vDash (x_i = y) \rightarrow (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$

T7.9 $\vDash (x_i = y) \rightarrow (\mathcal{R}^n x_1 \ldots x_i \ldots x_n \rightarrow \mathcal{R}^n x_1 \ldots y \ldots x_n)$

a. $\exists x Px \nvDash Pa$

*b. $\nvDash f^1 g^1 x = g^1 f^1 x$

c. $\exists x Fx, \exists y Gy \nvDash \exists z(Fz \wedge Gz)$

d. $\forall x \exists y Axy \nvDash \exists y \forall x Axy$

e. $\forall x \exists y(Ay \rightarrow Bx) \nvDash \forall x(\exists y Ay \rightarrow Bx)$

*E7.16. Provide demonstrations for (simplified versions of) T7.7 - T7.9 in the nonformalized style. Hint: You may or may not decide that a formalized derivation would be helpful. Challenge: can you show the theorems in their general form?

E7.17. Show that $\mathsf{N}[\forall x \forall y(Sx = Sy \to x = y)] = \mathsf{T}$, and so complete the demonstration that $\forall x \forall y(Sx = Sy \to x = y) \not\vDash \exists x(Sx = \emptyset)$. You may simply assert that $\mathsf{N}[\exists x(Sx = \emptyset)] \neq \mathsf{T}$ with justification, "from the text."

E7.18. Suppose we want to show that $\forall x \exists y Rxy$, $\forall x \exists y Ryx$, $\forall x \forall y \forall z((Rxy \wedge Ryz) \to Rxz) \not\vDash \exists x Rxx$.

*a. Explain why no interpretation with a finite universe will do.

b. Explain why the standard interpretation N with $\mathsf{U} = \{0, 1, 2 \ldots\}$ and $\mathsf{N}[R] = \{\langle \mathsf{m}, \mathsf{n} \rangle \mid m < n\}$ will not do.

c. Find an appropriate interpretation and use it to show that $\forall x \exists y Rxy$, $\forall x \exists y Ryx$, $\forall x \forall y \forall z((Rxy \wedge Ryz) \to Rxz) \not\vDash \exists x Rxx$.

E7.19. Here is an interpretation to show $\not\vDash \exists x \forall y[(Axy \wedge \sim Axy) \to (Axx \leftrightarrow Ayy)]$.

$\mathsf{U} = \{1, 2, 3 \ldots\}$

$\mathsf{I}[A] = \{\langle \mathsf{m}, \mathsf{n} \rangle \mid \mathsf{m} \leq \mathsf{n}$ and m is odd, or $\mathsf{m} < \mathsf{n}$ and m is even$\}$

So $\mathsf{I}[A]$ has members,

$\langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle \ldots \qquad \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 2, 5 \rangle \ldots$

$\langle 3, 3 \rangle, \langle 3, 4 \rangle, \langle 3, 5 \rangle \ldots \qquad \langle 4, 5 \rangle, \langle 4, 6 \rangle, \langle 4, 7 \rangle \ldots$

and so forth. Try to understand *why* this works, and why $\leq$ or $<$ will not work by themselves. Then see if you can find an interpretation where U has $\leq$ four members, and use your interpretation to demonstrate that $\not\vDash \exists x \forall y[(Axy \wedge \sim Axy) \to (Axx \leftrightarrow Ayy)]$.

E7.20. Consider $\mathscr{L}_{NT}$ as in chapter 6 (p. 301) with just constant $\emptyset$, the function symbols S, $+$ and $\times$, and the relation symbol $=$ along with the axioms of Robinson Arithmetic as in the Robinson and Peano reference on p. 314. Then (i) use the standard interpretation N to show that $\mathsf{Q} \not\vDash \sim\forall x[(\emptyset \times x) = \emptyset]$ and $\mathsf{Q} \not\vDash \sim\forall x \forall y[(x \times y) = (y \times x)]$. And (ii) take a nonstandard interpretation that has $\mathsf{U} = \{0, 1, 2 \ldots, a\}$ for some object a that is not a number; assign 0 to $\emptyset$ in the usual way. Then set,

S	
i	$i+1$
a	a

$+$	j	a
i	$i+j$	a
a	a	a

$\times$	0	$j \neq 0$	a
0	0	0	a
$i \neq 0$	0	$i \times j$	a
a	0	a	a

Use this interpretation to show Q $\not\models \forall x[(\emptyset \times x) = \emptyset]$ and Q $\not\models \forall x \forall y[(x \times y) = (y \times x)]$. This result, together with T10.3 according to which our derivation system is *sound* is sufficient to show that Robinson Arithmetic is not (negation) *complete* — there are sentences $\mathcal{P}$ of $\mathcal{L}_{\text{NT}}$ such that Q proves neither $\mathcal{P}$ nor $\sim\mathcal{P}$.

E7.21. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples of your own construction (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The difference between satisfaction and truth.

b. The definitions SF(r) and SF($\forall$).

c. The way your reasoning works. For this you can provide an example of some reasonably complex but clean bits of reasoning, (a) for validity, and (b) for invalidity. Then explain to Hannah how your reasoning works. That is, provide her a commentary on what you have done, so that she could understand.

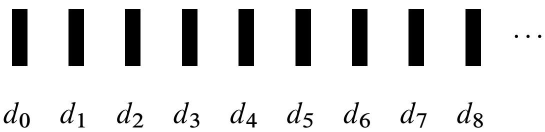
Chapter 8

Mathematical Induction

In chapter 1, (p. 11), we distinguished *deductive* from *inductive* arguments. As described there, in a deductive argument, conclusions are supposed to be *guaranteed* by premises. In an inductive argument, conclusions are merely made probable or plausible. Typical cases of inductive arguments involve generalization from cases. Thus, for example, one might reason from the premise that every crow we have ever seen is black, to the conclusion that all crows are black. The premise does not *guarantee* the conclusion, but it does give it some probability or plausibility. Similarly, mathematical induction involves a sort of generalization. But mathematical induction is a *deductive* argument form. The conclusion of a valid argument by mathematical induction is *guaranteed* by its premises. So mathematical induction is to be distinguished from the sort of induction described in chapter 1. In this chapter, I begin with a general characterization of mathematical induction, and turn to a series of exmples. Some of the examples will matter for things to come. But the primary aim is to gain facility with this crucial argument form.

8.1 General Characterization

Arguments by mathematical induction apply to objects that are arranged in *series*. The conclusion of an argument by mathematical induction is that all the elements of the series are of a certain sort. For cases with which we will be concerned, the elements of a series are ordered by integers: there is an initial member member, one after that, and so forth (we may thus think of a series as a *function* from the integers to the members). Consider, for example, a series of dominoes.



This series is ordered spatially. d_0 is the initial domino, d_1 the next, and so forth. Alternatively, we might think of the series as defined by a function $\mathcal{D}$ from the natural numbers to the dominoes, with $\mathcal{D}(0) = d_0$, $\mathcal{D}(1) = d_1$ and so forth — where this ordering is merely *exhibited* by the spatial arrangement.

Suppose we are interested in showing that *all* the dominoes fall, and consider the following two claims:

(i) the first domino falls

(ii) for any domino, if all the ones prior to it fall, then it falls.

By itself, (i) does not tell us that all the dominoes fall. For all we know, there might be some flaw in the series so that for some $j < k$, d_j falls, but d_k does not. Perhaps the space between d_{k-1} and d_k is too large. In this case, under ordinary circumstances, neither d_k nor any of the dominoes after it fall. (ii) tells us that there is no such flaw in the series — if all the dominoes up to d_k fall, then d_k falls. But (ii) is not, by itself, sufficient for the conclusion that all the dominoes fall. From the fact that the dominoes are so arranged, it does not follow that *any* of the dominoes fall. Perhaps you do the arrangement, and are so impressed with your work, that you leave the setup forever as a memorial!

However, given both (i) and (ii), it is safe to conclude that all the dominoes fall. There are a couple of ways to see this. First, we can reason from one domino to the next. By (i), the first domino falls. This means that all the dominoes prior to the second domino fall. So by (ii), the second falls. But this means all the dominoes prior to the third fall. So by (ii), the third falls. So all the dominoes prior to the fourth fall. And so forth. Thus we reach the conclusion that each domino falls. So all the dominoes fall. Here is another way to make the point: Suppose not every member of the series falls. Then there must be some *least* member d_a of the series which does not fall. d_a cannot be the first member of the series, since by (i) the first member of the series falls. And since d_a is the least member of the series which does not fall, all the members of the series prior to it *do* fall! So by (ii), d_a falls. This is impossible; reject the assumption: every member of the series falls.

Suppose we have some reason for accepting (i) that the first domino falls — perhaps you push it with your finger. Suppose further, that we have some "special reason" for moving from the premise that all the dominoes prior to an arbitrary d_k

fall, to the conclusion that d_k falls — perhaps the setup only gets better and better as the series continues, and the builder gains experience. Then we might attempt to show that all the dominoes fall as follows.

	a.	d_0 falls	prem (d_0 particular)
	b.	| all the dominoes prior to d_k fall	assp (d_k arbitrary)
		| ⋮	
(A)	c.	| d_k falls	"special reason"
	d.	if all the dominoes prior to d_k fall, then d_k falls	b-c cnd
	e.	for any domino, if all the dominoes prior to it fall, then it falls	d unv
	f.	every domino falls	a,e induction

(a) is just (i); d_0 falls because you push it. (e) is (ii); to get this, we reason from the assumption at (b), and the "special reason," to the conclusion that d_k falls, and then move to (e) by cnd and unv. The conclusion that every domino falls follows from (a) and (e) by mathematical induction. This is in fact how we reason. However, all the moves are automatic once we complete the subderivation — the moves by cnd to get (d), by unv to get (e), and by mathematical induction to get (f) are automatic once we reach (c). In practice, then, those steps are usually left implicit and omitted. Having gotten (a) and, from the assumption that all the dominoes prior to d_k fall, reached the conclusion that d_k falls, we move directly to the conclusion that all the dominoes fall.

Thus we arrive at a general form for arguments by mathematical induction. Suppose we want to show that $\mathscr{P}$ holds for each member of some series. Then an argument from mathematical induction goes as follows.

(B) *Basis:* Show that $\mathscr{P}$ holds for the first member of the series.

Assp: Assume, for arbitrary k, that $\mathscr{P}$ holds for every member of the series prior to the kth member.

Show: Show that $\mathscr{P}$ holds for the kth member.

Indct: Conclude that $\mathscr{P}$ holds for every member of the series.

In the domino case, for the *basis* we show (i). At the *assp* (assumption) step, we assume that all the dominoes prior to d_k fall. In the *show* step, we would complete the subderivation with the conclusion that domino d_k falls. From this, moves by cnd, to the conditional statement, and by unv to its generalization, are omitted, and we move directly to the conclusion that all the dominoes fall. Notice that the assumption is nothing more than a standard assumption for the (suppressed) application of cnd.

Perhaps the "special reason" is too special, and it is not clear how we might generally reason from the assumption that some $\mathscr{P}$ holds for every member of a series

prior to the kth, to the conclusion that it holds for the kth. For our purposes, the key is that such reasoning is possible in contexts characterized by *recursive definitions*. As we have seen, a recursive definition always moves from the parts to the whole. There are some *basic* elements, and some rules for combining elements to form further elements. In general, it is a fallacy (the fallacy of *composition*) to move directly from characteristics of parts, to characteristics of a whole. From the fact that the bricks are small, it does not follow that a building made from them is small. But there are cases where facts about parts, together with the way they are arranged, are sufficient for conclusions about wholes. If the bricks are hard, it may be that the building is hard. And similarly with recursive definitions.

To see how this works, let us turn to another example. We show that every *term* of a certain language has an odd number of symbols. Recall that the recursive definition TR tells us how terms are formed from others. Variables and constants are terms; and if h^n is a n-place function symbol and $t_1 \dots t_n$ are n terms, then $h^n t_1 \dots t_n$ is a term. On tree diagrams, across the top are variables and constants — terms with no function symbols; in the next row are terms constructed out of them, and for any $n > 1$, terms in row n are constructed out of terms from earlier rows. Let this *series* of rows be our series for mathematical induction. Every term must appear in some row of a tree. We consider a series whose first element consists of terms which appear in the top row of a tree, whose second element consists of terms which appear in the second, and so forth. Let $\mathscr{L}_t$ be a language with variables and constants as usual, but just two function symbols, a two-place function symbol f^2 and a four-place function symbol g^4. We show, by induction on the rows in which terms appear, that the total number of symbols in any term t of this language is odd. Here is the argument:

(C) *Basis:* If t appears in a top row (row zero), then it is a variable or a constant; in this case, t consists of just one variable or constant symbol; so the total number of symbols in t is odd.

Assp: For any i such that $0 \leq i < k$, the total number of symbols in any t appearing in row i is odd.

Show: The total number of symbols in any t appearing in row k is odd.

If t appears in row k, then it is of the form $f^2 t_1 t_2$ or $g^4 t_1 t_2 t_3 t_4$ where $t_1 \dots t_4$ appear in rows prior to k. So there are two cases.

(f) Suppose t is $f^2 t_1 t_2$. Let a be the total number of symbols in t_1 and b be the total number of symbols in t_2; then the total number of symbols in t is $(a + b) + 1$: all the symbols in t_1, all the symbols in t_2, plus the symbol f^2. Since t_1 and t_2 each appear in rows prior to k, by assumption, both a and b are odd. But the sum of two odds is an even,

and the sum of an even plus one is odd; so $(a + b) + 1$ is odd; so the total number of symbols in t is odd.

(g) Suppose t is $g^4 t_1 t_2 t_3 t_4$. Let a be the total number of symbols in t_1, b be the total number of symbols in t_2, c be the total number of symbols in t_3 and d be the total number of symbols in t_4; then the total number of symbols in t is $[(a + b) + (c + d)] + 1$. Since $t_1 \dots t_4$ each appear in rows prior to k, by assumption a, b, c and d are all odd. But the sum of two odds is an even; the sum of two evens is an even, and the sum of an even plus one is odd; so $[(a + b) + (c + d)] + 1$ is odd; so the total number of symbols in t is odd.

In either case, then, if t appears in row k, the total number of symbols in t is odd.

Indct: For any term t in $\mathscr{L}_t$, the total number of symbols in t is odd.

Notice that this argument is *entirely structured by the recursive definition for terms*. The definition TR includes clauses (v) and (c) for terms that appear in the top row. In the basis stage, we show that all such terms consist of an odd number of symbols. Then, for (suppressed) application of cnd and gen we assume that all terms prior to an arbitrary row k have an odd number of symbols. The show line simply announces what we plan to do. The sentence after derives directly from clause (f) of TR: In this case, there are just two ways to construct terms out of other terms. If $f^2 t_1 t_2$ appears in row k, t_1 and t_2 must appear in previous rows. So, by the assumption, they have an odd number of symbols. And similarly for $g^4 t_1 t_2 t_3 t_4$. In the reasoning for the show stage we demonstrate that, either way, if the total number of symbols in the parts are odd, then the total number of symbols in the whole is odd. It follows that every term in this language $\mathscr{L}_t$ consists of an odd number of symbols.

Returning to the domino analogy, the basis is like (i), where we show that the first member of the series falls — terms appearing in the top row always have an odd number of symbols. Then, for arbitrary k, we assume that all the members of the series prior to the kth fall — that terms appearing in rows prior to the kth always have an odd number of symbols. We then reason that, given this, the kth member falls — terms constructed out of others which, by assumption have an odd number of symbols, must themselves have an odd number of symbols. From this, (ii) follows by cnd and unv, and the general conclusion by mathematical induction.

The argument works for the same reasons as before: Insofar as a variable or constant is regarded as a single element of the vocabulary, it is automatic that variables and constants have an odd number of symbols. Given this, where function symbols

are also regarded as single elements of the vocabulary, expressions in the next row of a tree, as f^2xc, or g^4xycz, must have an odd number of symbols — one function symbol, plus two or four variables and constants. But if terms from the first and second rows of a tree have an odd number of symbols, by reasoning from the show step, terms constructed out of *them* must have an odd number of symbols as well. And so forth. Alternatively, suppose some terms in $\mathscr{L}_t$ have an even number of symbols; then there must be a least row a where such terms appear. From the basis, this row a is not the first. But since a is the least row at which terms have an even number of symbols, terms at all the earlier rows must have an odd number of symbols. But then, by reasoning as in the show step, terms at row a have an odd number of symbols. Reject the assumption, no terms in $\mathscr{L}_t$ have an even number of symbols.

In practice, for this sort of case, it is common to reason, not based on the row in which a term appears, but on the number of function symbols in the term. This differs in detail, but not in effect, from what we have done. In our trees, it may be that a term in the third row, combining one from the first and one from the second, has two function symbols, as f^2xf^2ab, or it may be that a term in the third row, combining ones from the second, has three function symbols, as $f^2f^2xyf^2ab$, or five, as $g^4f^2xyf^2abf^2zwf^2cd$, and so forth. However, it remains that the total number of function symbols in each of some terms $\mathscr{s}_1 \dots \mathscr{s}_n$ is fewer than the total number of function symbols in $\mathscr{h}^n\mathscr{s}_1 \dots \mathscr{s}_n$; for the latter includes all the function symbols in $\mathscr{s}_1 \dots \mathscr{s}_n$ plus $\mathscr{h}^n$. Thus we may consider the series: terms with no function symbols, terms with one function symbol, and so forth — and be sure that for any $n > 0$, terms at stage n are constructed of ones before. Here is a sketch of the argument modified along these lines.

(D) *Basis:* If t has no function symbols, then it is a variable or a constant; in this case, t consists of just the one variable or constant symbol; so the total number of symbols in t is odd.

Assp: For any i such that $0 \leq i < k$, the total number of symbols in t with i function symbols is odd.

Show: The total number of symbols in t with k function symbols is odd.

If t has k function symbols, then it is of the form $f^2t_1t_2$ or $g^4t_1t_2t_3t_4$ where $t_1 \dots t_4$ have less than k function symbols. So there are two cases.

(f) Suppose t is $f^2t_1t_2$. [As before…] the total number of symbols in t is odd.

(*g*) Suppose t is $g^4 t_1 t_2 t_3 t_4$. [As before...] the total number of symbols in t is odd.

In either case, then, if t has k function symbols, then the total number of symbols in t is odd.

Indct: For any term t in $\mathcal{L}_t$, the total number of symbols in t is odd.

Here is the key point: If $f^2 t_1 t_2$ has k function symbols, the total number of function symbols in t_1 and t_2 combined has to be $k - 1$; and since the number of function symbols in t_1 and in t_2 must individually be less than or equal to the combined total, the number of function symbols in t_1 and the number of function symbols in t_2 must also be less than k. And similarly for $g^4 t_1 t_2 t_3 t_4$. That is why the inductive assumption applies to $t_1 \ldots t_4$, and reasoning in the cases can proceed as before. If you find this confusing, you might picture our trees "regimented" so that rows correspond to the number of function symbols. Then this reasoning is no different than before.

8.2 Preliminary Examples

Let us turn now to a series of examples, meant to illustrate mathematical induction in a variety of contexts. Some of the examples have to do with our subject matter. But some do not. For now, the primary aim is to gain facility with the argument form. As you work through the cases, think about *why* the induction works. At first, examples may be difficult to follow. But they should be more clear by the end.

8.2.1 Case

First, a case where the conclusion may seem too obvious even to merit argument. We show that, any (official) formula $\mathcal{P}$ of a quantificational language has an equal number of left and right parentheses. Again, the relevant definition FR is recursive. Its basis clause specifies formulas without operator symbols; these occur across the top row of our trees. FR then includes clauses which say how complex formulas are constructed out of those that are less complex. We take as our series, formulas with no operator symbols, formulas with one operator symbol, and so forth; thus the argument is by induction on the *number of operator symbols*. As in the above case with terms, this orders formulas so that we can use facts from the recursive definition in our reasoning. Let us say $\mathrm{L}(\mathcal{P})$ is the number of left parentheses in $\mathcal{P}$, and $\mathrm{R}(\mathcal{P})$ is the number of right parentheses in $\mathcal{P}$. Our goal is to show that for any formula $\mathcal{P}$, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Induction Schemes

Schemes for mathematical induction sometimes appear in different forms. But for our purposes, these amount to the same thing. Suppose a series of objects, and consider the following.

I.
(a) Show that $\mathcal{P}$ holds for the first member
(b) Assume that $\mathcal{P}$ holds for members $< k$
(c) Show $\mathcal{P}$ holds for member k
(d) Conclude $\mathcal{P}$ holds for every member

This is the form as we have seen it.

II.
(a) Show that $\mathcal{P}$ holds for the first member
(b) Assume that $\mathcal{P}$ holds for members $\leq j$
(c) Show $\mathcal{P}$ holds for member $j + 1$
(d) Conclude $\mathcal{P}$ holds for every member

This comes to the same thing if we think of j as $k - 1$. Then $\mathcal{P}$ holds for members $\leq j$ just in case it holds for members $< k$.

III.
(a) Show that $\mathcal{Q}$ holds for the first member
(b) Assume that $\mathcal{Q}$ holds for member j
(c) Show $\mathcal{Q}$ holds for member $j + 1$
(d) Conclude $\mathcal{Q}$ holds for every member

This comes to the same thing if we think of j as $k - 1$ and $\mathcal{Q}$ as the proposition that $\mathcal{P}$ holds for members $\leq j$.

And similarly the other forms follow from ours. So, though in a given context, one form may be more convenient than another, the forms are equivalent — or at least they are equivalent for sequences corresponding to the natural numbers.

Where ω is the first infinite ordinal, there is no ordinal α such that $\alpha + 1 = \omega$. So for a sequence ordered by these ordinals, our assumption that $\mathcal{P}$ holds for all the members $< k$ might hold though there *is* no $j = k - 1$ as in the second and third cases. So the equivalence between the forms breaks down for series that are so ordered. We do not need to worry about infinite ordinals, as our concerns will be restricted to series ordered by the integers.

Our form of induction (I) is known as "Strong Induction," for its relatively strong inductive assumption, and the third as "Weak." The second is a sometimes-encountered blend of the other two.

(E) *Basis:* If $\mathcal{P}$ has no operator symbols, then $\mathcal{P}$ is a sentence letter $\mathcal{S}$ or an atomic $\mathcal{R}^n t_1 \dots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$. In either case, $\mathcal{P}$ has no parentheses. So $\mathrm{L}(\mathcal{P}) = 0$ and $\mathrm{R}(\mathcal{P}) = 0$; so $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Assp: For any i such that $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols, then $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Show: For every $\mathcal{P}$ with k operator symbols, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $(\mathcal{A} \rightarrow \mathcal{B})$, or $\forall x\mathcal{A}$ for variable x and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. Then $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A})$ and $\mathrm{R}(\mathcal{P}) = \mathrm{R}(\mathcal{A})$. But by assumption $\mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A})$; so $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

($\rightarrow$) Suppose $\mathcal{P}$ is $(\mathcal{A} \rightarrow \mathcal{B})$. Then $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A}) + \mathrm{L}(\mathcal{B}) + 1$ and $\mathrm{R}(\mathcal{P}) = \mathrm{R}(\mathcal{A}) + \mathrm{R}(\mathcal{B}) + 1$. But by assumption $\mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A})$, and $\mathrm{L}(\mathcal{B}) = \mathrm{R}(\mathcal{B})$; so the sums are the same, and $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$

($\forall$) Suppose $\mathcal{P}$ is $\forall x\mathcal{A}$. Then as in the case for ($\sim$), $\mathrm{L}(\mathcal{P}) = \mathrm{L}(\mathcal{A})$ and $\mathrm{R}(\mathcal{P}) = \mathrm{R}(\mathcal{A})$. But by assumption $\mathrm{L}(\mathcal{A}) = \mathrm{R}(\mathcal{A})$; so $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

If $\mathcal{P}$ has k operator symbols, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

Indct: For any formula $\mathcal{P}$, $\mathrm{L}(\mathcal{P}) = \mathrm{R}(\mathcal{P})$.

No doubt, you already knew that the numbers of left and right parentheses match. But, presumably, you knew it by reasoning of *this very sort*. Atomic formulas have no parentheses; after that, parentheses are always added in pairs; so, no matter how complex a formula may be, there is never a left parenthesis without a right to match. Reasoning by mathematical induction may thus seem perfectly natural! All we have done is to make explicit the various stages that are required to reach the conclusion. But it is important to make the stages explicit, for in many cases results are not so obvious. Here are some closely related problems.

*E8.1. For any (official) formula $\mathcal{P}$ of a quantificational language, where $\mathrm{A}(\mathcal{P})$ is the number of its atomic formulas, and $\mathrm{C}(\mathcal{P})$ is the number of its arrow symbols, show that $\mathrm{A}(\mathcal{P}) = \mathrm{C}(\mathcal{P}) + 1$. Hint: Argue by induction on the number of operator symbols in $\mathcal{P}$. For the basis, when $\mathcal{P}$ has no operator symbols, it is an atomic, so that $\mathrm{A}(\mathcal{P}) = 1$ and $\mathrm{C}(\mathcal{P}) = 0$. Then, as above, you will have cases for $\sim$, $\rightarrow$, and $\forall$. The hardest case is when $\mathcal{P}$ is of the form $(\mathcal{A} \rightarrow \mathcal{B})$.

E8.2. Consider now expressions which allow abbreviations ($\vee$), ($\wedge$), ($\leftrightarrow$), and ($\exists$). Where $\mathrm{A}(\mathcal{P})$ is the number of atomic formulas in $\mathcal{P}$ and $\mathrm{B}(\mathcal{P})$ is the number of its binary operators, show that $\mathrm{A}(\mathcal{P}) = \mathrm{B}(\mathcal{P}) + 1$. Hint: now you have seven cases: ($\sim$), ($\rightarrow$), and ($\forall$) as before, but also cases for ($\vee$), ($\wedge$), ($\leftrightarrow$), and ($\exists$). This suggests the beauty of reasoning just about the minimal language!

8.2.2 Case

Mathematical induction is so-called because many applications occur in mathematics. It will be helpful to have a couple of examples of this sort. These should be illuminating — at least if you do not get bogged down in the details of the arithmetic! The series of odd integers is 1, 3, 5, 7 ... where the nth odd integer is $2n - 1$. (The nth even integer is $2n$; to find the nth odd, go to the even just above it, and come down one.) Let $\mathrm{S}(n)$ be the sum of the first n odd integers. So $\mathrm{S}(1) = 1$, $\mathrm{S}(2) = 1 + 3 = 4$, $\mathrm{S}(3) = 1 + 3 + 5 = 9$, $\mathrm{S}(4) = 1 + 3 + 5 + 7 = 16$ and, in general,

$$\mathrm{S}(n) = 1 + 3 + \ldots + (2n - 1)$$

We consider the series of these sums, $\mathrm{S}(1)$, $\mathrm{S}(2)$, and so forth, and show that, for any $n \geq 1$, $\mathrm{S}(n) = n^2$. The key to our argument is the realization that the sum of all the odd numbers up to the nth odd number is equal to the sum of all the odd numbers up to the $(n - 1)$th odd number plus the nth odd number. That is, since the nth odd number is $2n - 1$, $\mathrm{S}(n) = \mathrm{S}(n - 1) + (2n - 1)$. We argue by induction on the series of sums.

(F) *Basis:* If $n = 1$ then $\mathrm{S}(n) = 1$ and $n^2 = 1$; so $\mathrm{S}(n) = n^2$.

Assp: For any i, $1 \leq i < k$, $\mathrm{S}(i) = i^2$.

Show: $\mathrm{S}(k) = k^2$. As above, $\mathrm{S}(k) = \mathrm{S}(k-1) + (2k-1)$. But since $k-1 < k$, by the inductive assumption, $\mathrm{S}(k-1) = (k-1)^2$; so $\mathrm{S}(k) = (k-1)^2 + (2k - 1) = (k^2 - 2k + 1) + (2k - 1) = k^2$. So $\mathrm{S}(k) = k^2$.

Indct: For any n, $\mathrm{S}(n) = n^2$.

As is often the case in mathematical arguments, the kth element is completely determined by the one before; so we do not need to consider any more than this one way that elements at stage k are determined by those at earlier stages.[1] Surely this is

[1]Thus arguments by induction in arithmetic and geometry are often conveniently cast according to the third "weak" induction scheme from induction schemes on p. 384. But, as above, our standard scheme applies as well.

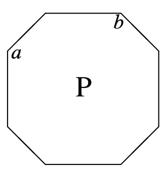
an interesting result — though you might have wondered about it after testing initial cases, we have a demonstration that it holds for every n.

*E8.3. Let S(n) be the sum of the first n even integers; that is S(n) $= 2+4+\ldots+2n$. So S(1) $= 2$, S(2) $= 2+4 = 6$, S(3) $= 2+4+6 = 12$, and so forth. Show, by mathematical induction, that for any $n \geq 1$, S(n) $= n(n+1)$.

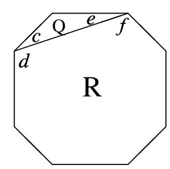
E8.4. Let S(n) be the sum of the first n integers; that is S(n) $= 1+2+3+\ldots+n$. So S(1) $= 1$, S(2) $= 1+2 = 3$, S(3) $= 1+2+3 = 6$, and so forth. Show, by mathematical induction, that for any $n \geq 1$, S(n) $= n(n+1)/2$.

8.2.3 Case

Now a case from geometry. Where a polygon is *convex* iff each of its interior angles is less than 180°, we show that the sum of the interior angles in any convex polygon with n sides, S(P) $= (n-2)180°$. Let us consider polygons with three sides, polygons with four sides, polygons with five sides, and so forth. The key is that any n-sided polygon may be regarded as one with $n-1$ sides combined with a triangle. Thus given an n-sided polygon P,

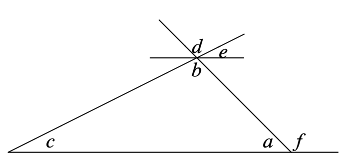


Construct a line connecting opposite ends of a pair of adjacent sides.



The result is a triangle Q and a figure R with n$-$1 sides, where a $=$ c$+$d and b $=$ e$+$f. The sum of the interior angles of P is the same as the sum of the interior angles of Q plus the sum of the interior angles of R. Once we realize this, our argument by mathematical induction is straightforward. For any convex n-sided polygon P, we show that the sum of the interior angles of P, S(P) $=$ (n $-$ 2)180°. The argument is by induction on the number n of sides of the polygon.

(G) *Basis:* If $n = 3$, then P is a triangle; but by reasoning as follows,



By definition, a $+$ f $=$ 180°; but b $=$ d and if the horizontal lines are parallel, c $=$ e and d $+$ e $=$ f; so a $+$ (b $+$ c) $=$ a $+$ (d $+$ e) $=$ a $+$ f $=$ 180°.

the sum of the angles in a triangle is 180°. So S(P) $=$ 180. But $(3 - 2)180 = 180$. So S(P) $= (n - 2)180$.

Assp: For any i, $3 \leq i < k$, every P with i sides has S(P) $= (i - 2)180$.

Show: For every P with k sides, S(P) $= (k - 2)180$.

If P has k sides, then for some triangle Q and polygon R with $k-1$ sides, S(P) $=$ S(Q) $+$ S(R). Q is a triangle, so S(Q) $= 180$. Since $k - 1 < k$, the inductive assumption applies to R; so S(R) $= ((k - 1) - 2)180$. So S(P) $= 180 + ((k - 1) - 2)180 = (1 + k - 1 - 2)180 = (k - 2)180$. So S(P) $= (k - 2)180$.

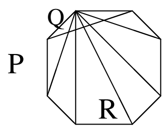
Indct: For any n-sided polygon P, S(P) $= (n - 2)180$.

Perhaps reasoning in the basis brings back good (or bad) memories of high school geometry! But you do not have to worry about that. In this case, the sum of the angles of a figure with n sides is completely determined once we are given the sum of the angles for a figure with $n - 1$ sides. So we do not need to consider any more than this one way that elements at stage k are determined by those at earlier stages.

It is worth noting however that we do not have to see a k-sided polygon as composed of a triangle and a figure with $k - 1$ sides. For consider *any* diagonal of a k-sided polygon; it divides the figure into two, each with $< k$ sides. So the inductive assumption applies to each figure. So we might reason about the angles of a k-sided figure as the sum of angles of these arbitrary parts, as in the exercise that follows.

*E8.5. Using the fact that any diagonal of a k-sided polygon divides it into two polygons with $< k$ sides, show by mathematical induction that the sum of the interior angles of any convex polygon P, S(P) $= (n - 2)180$. Hint: If a figure has k sides, then for some a such that both a and $k - a$ are at least two (> 1), a diagonal divides it into a figure Q with $a + 1$ sides (a sides from P, plus the diagonal), and a figure R with $(k - a) + 1$ sides (the remaining sides from P, plus the diagonal). From $a > 1$, $k + a > k + 1$ so that $k > k - a + 1$; and from $k - a > 1$, $k > a + 1$. So the inductive assumption applies to both Q and R.

E8.6. Where P is a convex polygon with n sides, and D(P) is the number of its diagonals (where a *diagonal* is a line from one vertex to another that is not a side), show by mathematical induction that any P with $n \geq 3$ sides is such that D(P) $= n(n - 3)/2$.



Hint: When you add a triangle to a convex figure to form a new convex figure with k sides, the diagonals are all the diagonals you had before, plus the base of the triangle, plus $k-3$ lines from vertices not belonging to the triangle to the apex of the triangle.

Also, in case your algebra is rusty, $(k-1)(k-4) = k^2 - 5k + 4$.

8.2.4 Case

Finally we take up a couple of cases of real interest for our purposes — though we limit consideration just to sentential forms. We have seen cases structured by the recursive definitions TR and FR. Here is one that uses ST. Say a formula is in *normal* form iff its only operators are $\vee$, $\wedge$, and $\sim$, and the only instances of $\sim$ are immediately prefixed to atomics (of course, any normal form is an abbreviation of a formula whose only operators are $\rightarrow$ and $\sim$). Where $\mathcal{P}$ is a normal form, let $\mathcal{P}'$ be like $\mathcal{P}$ except that $\vee$ and $\wedge$ are interchanged and, for a sentence letter $\mathcal{S}$, $\mathcal{S}$ and $\sim\mathcal{S}$ are interchanged. Thus, for example, if $\mathcal{P}$ is an atomic A, then $\mathcal{P}'$ is $\sim A$, if $\mathcal{P}$ is $(A \vee (\sim B \wedge C))$, then $\mathcal{P}'$ is $(\sim A \wedge (B \vee \sim C))$. We show that if $\mathcal{P}$ is in normal form, then $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. Thus, for the case we have just seen,

$$\mathsf{I}[\sim(A \vee (\sim B \wedge C))] = \mathsf{T} \quad \text{iff} \quad \mathsf{I}[(\sim A \wedge (B \vee \sim C))] = \mathsf{T}$$

So the result works like a generalized semantic version of DeM in combination with DN: When you push a negation into a normal form, $\wedge$ flips to $\vee$, $\vee$ flips to $\wedge$, and atomics switch between $\mathcal{S}$ and $\sim\mathcal{S}$.

Our argument is by induction on the number of operators in a formula $\mathcal{P}$. Let $\mathcal{P}$ be any normal form.

(H) *Basis:* If $\mathcal{P}$ has no operators, then $\mathcal{P}$ is an atomic $\mathcal{S}$; so $\sim\mathcal{P} = \sim\mathcal{S}$ and $\mathcal{P}' = \sim\mathcal{S}$; so $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

Assp: For any i, $0 \leq i < k$, any $\mathcal{P}$ in normal form with i operator symbols is such that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

Show: Any $\mathcal{P}$ in normal form with k operator symbols is such that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

If $\mathcal{P}$ is in normal form and has k operator symbols, then it is of the form $\sim\mathcal{S}$, $\mathcal{A} \vee \mathcal{B}$, or $\mathcal{A} \wedge \mathcal{B}$ where $\mathcal{S}$ is atomic and $\mathcal{A}$ and $\mathcal{B}$ are in normal form with less than k operator symbols. So there are three cases.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{S}$. Then $\sim\mathcal{P}$ is $\sim\sim\mathcal{S}$, and $\mathcal{P}'$ is $\mathcal{S}$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\sim\mathcal{S}] = \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\sim\mathcal{S}] \neq \mathsf{T}$; by ST($\sim$) again iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

($\vee$) Suppose $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$. Then $\sim\mathcal{P}$ is $\sim(\mathcal{A} \vee \mathcal{B})$, and $\mathcal{P}'$ is $\mathcal{A}' \wedge \mathcal{B}'$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim(\mathcal{A} \vee \mathcal{B})] = \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\mathcal{A} \vee \mathcal{B}] \neq \mathsf{T}$; by ST$'$($\vee$) iff $\mathsf{I}[\mathcal{A}] \neq \mathsf{T}$ and $\mathsf{I}[\mathcal{B}] \neq \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\sim\mathcal{A}] = \mathsf{T}$ and $\mathsf{I}[\sim\mathcal{B}] = \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}'] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B}'] = \mathsf{T}$; by ST$'$($\wedge$) iff $\mathsf{I}[\mathcal{A}' \wedge \mathcal{B}'] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. So $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

($\wedge$) Homework.

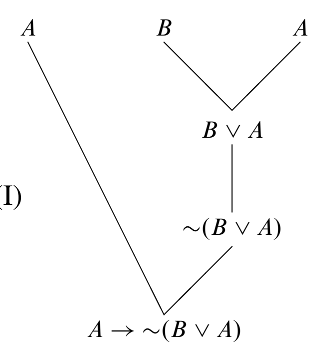
Every $\mathcal{P}$ with k operator symbols is such that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

Indct: Every $\mathcal{P}$ is such that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$.

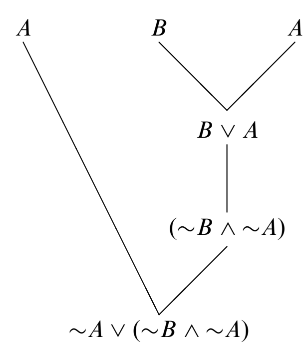
For the show step, it is important that $\mathcal{A}$ and $\mathcal{B}$ are in normal form. If they were not, then the inductive assumption, which applies only to formulas in normal form, would not apply to them. Similarly, it is important that $\mathcal{A}$ and $\mathcal{B}$ have $< k$ operators. If they did not, then the inductive assumption, which applies only to formulas with $< k$ operators, would not apply to them. The pattern here is typical: In the cases, we break down to parts to which the assumption applies, apply the assumption, and put the resultant parts back together. In the second case, we assert that if $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$, then $\mathcal{P}'$ is $\mathcal{A}' \wedge \mathcal{B}'$. Here $\mathcal{A}$ and $\mathcal{B}$ may be complex. We do the conversion on $\mathcal{P}$ iff we do the conversion on its main operator, and then do the conversion on its parts. And similarly for ($\wedge$). It is this which enables us to feed into the inductive assumption. Notice that it is convenient to cast reasoning in the "collapsed" biconditional style.

Where $\mathcal{P}$ is *any* form whose operators are $\sim$, $\vee$, $\wedge$, or $\to$, we now show that $\mathcal{P}$ is equivalent to a normal form. Consider a transform $\mathcal{P}^*$ defined as follows: For atomic $\mathcal{S}$, $\mathcal{S}^* = \mathcal{S}$; for arbitrary formulas $\mathcal{A}$ and $\mathcal{B}$ with just those operators, $(\mathcal{A} \vee \mathcal{B})^* = (\mathcal{A}^* \vee \mathcal{B}^*)$, $(\mathcal{A} \wedge \mathcal{B})^* = (\mathcal{A}^* \wedge \mathcal{B}^*)$, and with prime defined as above, $(\mathcal{A} \to \mathcal{B})^* = ([\mathcal{A}^*]' \vee \mathcal{B}^*)$, and $[\sim\mathcal{A}]^* = [\mathcal{A}^*]'$. To see how this works, consider how you would construct $\mathcal{P}^*$ on a tree.

(I)



For any $\mathcal{P}$ on the left, the corresponding $\mathcal{P}^*$ appears on the right



For the last line, A^* is A and $A^{*\prime}$ is $\sim A$. The star-transform, and the right-hand tree works very much like unabbreviating from subsection 2.1.3. The conversion of a complex formula depends on the conversion of its parts. So starting with the parts, we construct the star-transform of the whole, one component at a time. Observe that, at each stage of the right-hand tree, the result is a normal form.

We show by mathematical induction on the number of operators in $\mathcal{P}$ that $\mathcal{P}^*$ must be a normal form and that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$. For the argument it will be important, not only to use the inductive assumption, but also the result from above that for any $\mathcal{P}$ in normal form, $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. In order to apply this result, it will be crucial that every $\mathcal{P}^*$ is in normal form! Let $\mathcal{P}$ be any formula with just operators $\sim$, $\vee$, $\wedge$ and $\rightarrow$. Here is an outline of the argument, with parts left as homework.

T8.1. For any $\mathcal{P}$ whose operators are $\sim$, $\vee$, $\wedge$ and $\rightarrow$, $\mathcal{P}^*$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

Basis: If $\mathcal{P}$ is an atomic $\mathcal{S}$, then $\mathcal{P}^* = \mathcal{S}$. But an atomic $\mathcal{S}$ is in normal form; so $\mathcal{P}^*$ is in normal form; and since they are the same $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

Assp: For any i, $0 \leq i < k$ if $\mathcal{P}$ has i operator symbols, then $\mathcal{P}^*$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

Show: If $\mathcal{P}$ has k operator symbols, then $\mathcal{P}^*$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

If $\mathcal{P}$ has k operator symbols, then $\mathcal{P}$ is of the form $\sim\mathcal{A}$, $\mathcal{A}\vee\mathcal{B}$, $\mathcal{A}\wedge\mathcal{B}$, or $\mathcal{A} \rightarrow \mathcal{B}$ for formulas $\mathcal{A}$ and $\mathcal{B}$ with less than k operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. Then $\mathcal{P}^* = [\mathcal{A}^*]'$. By assumption $\mathcal{A}^*$ is in normal form; so since the prime operation converts a normal form to another normal form, $[\mathcal{A}^*]'$ is in normal form; so $\mathcal{P}^*$ is in normal form. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\mathcal{A}] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{A}] \neq \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}^*] \neq \mathsf{T}$; by ST($\sim$) iff $\mathsf{I}[\sim(\mathcal{A}^*)] = \mathsf{T}$; by assumption $\mathcal{A}^*$ is in normal form, so by our previous result, iff $\mathsf{I}[(\mathcal{A}^*)'] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$. So $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

($\wedge$) Homework.

($\vee$) Homework.

($\rightarrow$) Homework.

In any case, if $\mathcal{P}$ has k operator symbols, $\mathcal{P}^*$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

Indct: For any $\mathcal{P}$, $\mathcal{P}^*$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

The inductive assumption applies just to formulas with $< k$ operator symbols. So it applies just to formulas on the order of $\mathcal{A}$ and $\mathcal{B}$. The result from before applies to any formulas in normal form. So it applies to $\mathcal{A}^*$, once we have determined that $\mathcal{A}^*$ is in normal form.

E8.7. Complete induction (H) to show that every $\mathcal{P}$ in normal form is such that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}'] = \mathsf{T}$. You should set up the whole induction with statements for the basis, assumption and show parts. But then you may appeal to the text for parts already done, as the text appeals to homework. Hint: If $\mathcal{P} = (\mathcal{A} \wedge B)$ then $\mathcal{P}' = (\mathcal{A}' \vee \mathcal{B}')$.

E8.8. Complete T8.1 to show that any $\mathcal{P}$ with just operators $\sim$, $\vee$, $\wedge$ and $\rightarrow$ has a $\mathcal{P}^*$ in normal form such that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$. Again, you should set up the whole induction with statements for the basis, assumption and show parts. But then you may appeal to the text for parts already done, as the text appeals to homework.

E8.9. Show that for any $\mathcal{P}$ whose operators are $\sim$, $\vee$, $\wedge$ and $\rightarrow$, $\mathcal{P}^*$ is in normal form and $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^*$. Hint: the reasoning is parallel to the semantic case, but now about what you can *derive*. You will need results for both the prime and star.

E8.10. Let $\mathsf{I}[\mathcal{S}] = \mathsf{T}$ for every sentence letter $\mathcal{S}$. Where $\mathcal{P}$ is any sentential formula whose only operators are $\rightarrow$, $\wedge$, $\vee$ and $\leftrightarrow$, show by induction on the number of operators in $\mathcal{P}$ that $\mathsf{I}[\mathcal{P}] = \mathsf{T}$. Use this result to show that $\nvDash_s \sim\mathcal{P}$.

8.2.5 Case

Here is a result like one we will seek later for the quantificational case. It depends on the (recursive) notion of a *derivation*. Because of their relative simplicity, we will focus on axiomatic derivations. If we were working with "derivations" of the sort described in the diagram on p. 70, then we could reason by induction on the row in which a formula appears. Formulas in the top row result directly as axioms, those in the next row from ones before with MP; and so forth. Similarly, we could "regiment"

diagrams and proceed by induction on the number of applications of MP by which a formula is derived. But our official notion of an axiomatic derivation is not this; in an official axiomatic derivation, lines are ordered, where each line is either an axiom, a premise, or follows from previous lines by a rule. But this is sufficient for us to reason about one line of an axiomatic derivation based on ones that come before; that is, we reason by induction on the line number of a derivation. Say $\Gamma \vdash_{ADs} \mathcal{P}$ just in case there is a derivation of $\mathcal{P}$ in the sentential fragment of *AD*; that is, there is a derivation using just A1, A2, A3 and MP from definition AS. We show that if $\mathcal{P}$ is a theorem of *ADs*, then $\mathcal{P}$ is true on any sentential interpretation: if $\vdash_{ADs} \mathcal{P}$ then $\vDash_s \mathcal{P}$. Insofar as it applies where there are no premises, this result is known as *weak soundness*.

Suppose $\vdash_{ADs} \mathcal{P}$; then there is an *ADs* derivation $\langle \mathcal{A}_1, \mathcal{A}_2 \ldots \mathcal{A}_n \rangle$ of $\mathcal{P}$ from no premises, with $\mathcal{A}_n = \mathcal{P}$. By induction on the line numbers of this derivation, we show that for any j, $\vDash_s \mathcal{A}_j$. The case when $j = n$ is the desired result.

(J) *Basis:* Since $\langle \mathcal{A}_1, \mathcal{A}_2 \ldots \mathcal{A}_n \rangle$ is a derivation from no premises, $\mathcal{A}_1$ can only be an instance of A1, A2 or A3.

(A1) Say $\mathcal{A}_1$ is an instance of A1 and so of the form $\mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$. Suppose $\nvDash_s \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$; then by SV, there is an I such that $\mathsf{I}[\mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})] \neq \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q} \rightarrow \mathcal{P}] \neq \mathsf{T}$; from the latter, by ST($\rightarrow$), $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$. This is impossible; reject the assumption: $\vDash_s \mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P})$.

(A2) Similarly.

(A3) Similarly.

Assp: For any i, $1 \leq i < k$, $\vDash_s \mathcal{A}_i$.

Show: $\vDash_s \mathcal{A}_k$.

$\mathcal{A}_k$ is either an axiom or arises from previous lines by MP. If $\mathcal{A}_k$ is an axiom then, as in the basis, $\vDash_s A_k$. So suppose $\mathcal{A}_k$ arises from previous lines by MP. In this case, the picture is something like this:

a.	$\mathcal{B} \rightarrow \mathcal{C}$	
b.	$\mathcal{B}$	
k.	$\mathcal{C}$	a,b MP

where $a, b < k$ and $\mathcal{C}$ is $\mathcal{A}_k$. By assumption, $\vDash_s \mathcal{B}$ and $\vDash_s \mathcal{B} \rightarrow \mathcal{C}$. Suppose $\nvDash_s \mathcal{A}_k$; then $\nvDash_s \mathcal{C}$; so by SV there is some I such that $\mathsf{I}[\mathcal{C}] \neq \mathsf{T}$; let J be a particular interpretation of this sort; then $\mathsf{J}[\mathcal{C}] \neq \mathsf{T}$; but by SV, for any I, $\mathsf{I}[\mathcal{B}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B} \rightarrow \mathcal{C}] = \mathsf{T}$; so $\mathsf{J}[\mathcal{B}] = \mathsf{T}$ and $\mathsf{J}[\mathcal{B} \rightarrow \mathcal{C}] =$

T; from the latter, by ST($\rightarrow$), $\mathsf{J}[\mathcal{B}] \neq \mathsf{T}$ or $\mathsf{J}[\mathcal{C}] = \mathsf{T}$; so $\mathsf{J}[\mathcal{C}] = \mathsf{T}$. This is impossible; reject the assumption: $\vDash_s \mathcal{A}_k$.

Indct: For any line j of the derivation $\vDash_s A_j$.

We might have continued as above for (A2) and (A3). Alternatively, since we have already done the work, we might have appealed directly to T7.2s, T7.3s and T7.4s for (A1), (A2) and (A3) respectively. From the case when $\mathcal{A}_j = \mathcal{P}$ we have $\vDash_s \mathcal{P}$. This result is a precursor to one we will obtain in chapter 10. There, we will show *strong soundness* for the complete system *AD*, if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. This tells us that our derivation system can never lead us astray. There is no situation where a derivation moves from premises that are true, to a conclusion that is not. Still, what we have is interesting in its own right: It is a first connection between the syntactic notions associated with derivations, and the semantic notions of validity and truth.

E8.11. Let *A3* be like *A2* for exercise E3.4 (p. 79) except that the rule MP is stated entirely in $\sim$ and $\wedge$. Then the axiom and rule schemes are,

A3 A1. $\mathcal{P} \rightarrow (\mathcal{P} \wedge \mathcal{P})$

A2. $(\mathcal{P} \wedge \mathcal{Q}) \rightarrow \mathcal{P}$

A3. $(\mathcal{O} \rightarrow \mathcal{P}) \rightarrow [\sim(\mathcal{P} \wedge \mathcal{Q}) \rightarrow \sim(\mathcal{Q} \wedge \mathcal{O})]$

MP. $\sim(\mathcal{P} \wedge \sim\mathcal{Q}), \mathcal{P} \vDash \mathcal{Q}$

Show by mathematical induction that *A3* is weakly sound. That is show that if $\vdash_{A3} \mathcal{P}$ then $\vDash_s \mathcal{P}$.

E8.12. Modify your above argument to show that *A3* is strongly sound. That is, modify the argument to show that if $\Gamma \vdash_{A3} \mathcal{P}$ then $\Gamma \vDash_s \mathcal{P}$. You may appeal to reasoning from the previous problem where it is applicable. Hint: When premises are allowed, $\mathcal{A}_j$ is either an axiom, a premise, or arises by a rule. So there is one additional case in the basis; but that case is trivial – if all of the premises are true, and $\mathcal{A}_j$ is a premise, then $\mathcal{A}_j$ cannot be false. And your reasoning for the show will be modified; now the assumption gives you $\Gamma \vDash_s \mathcal{B} \rightarrow \mathcal{C}$ and $\Gamma \vDash_s \mathcal{B}$ and your goal is to show $\Gamma \vDash_s \mathcal{C}$.

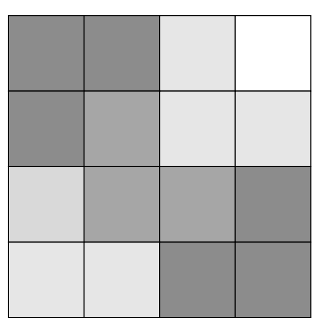
E8.13. Modify table T($\sim$) so that $\mathsf{I}[\sim\mathcal{P}] = \mathsf{F}$ both when $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{P}] = \mathsf{F}$; let table T($\rightarrow$) remain as before. Say a formula is *ideal* iff it is true on every

interpretation, given the revised tables. Show by mathematical induction that every consequence in *AD* of MP with A1 and A2 alone is ideal. Then by a table show that A3 is not ideal, and so that there is no derivation of A3 from A1 and A2 alone. Hint: your induction may be a simple modification of argument (J) from above.

E8.14. Where t is a term of $\mathscr{L}_q$, let $X(t)$ be the sum of all the superscripts in t and $Y(t)$ be the number of symbols in t. So, for example, if t is z, then $X(t) = 0$ and $Y(t) = 1$; if t is $g^1 f^2 cx$, then $X(t) = 3$ and $Y(t) = 4$. By induction on the number of function symbols in t, show that for any t in $\mathscr{L}_q$, $X(t) + 1 = Y(t)$.

E8.15. Show, by mathematical induction, that at a recent convention, the number of logicians who shook hands an odd number of times is even. Assume that 0 is even. Hints: Reason by induction on the number of handshakes at the convention. At any stage n, let O(n) be the number of people who have shaken hands an odd number of times. Your task is to show that for any n, O(n) is even. You will want to consider cases for what happens to O(n) when (i) someone who has already shaken hands an odd number of times shakes with someone who has shaken an odd number of times; (ii) someone who has already shaken hands an even number of times shakes with someone who has shaken an even number of times; and (iii) someone who has already shaken hands an odd number of times shakes with someone who has shaken an even number of times.

E8.16. For any $n \geq 1$, given a $2^n \times 2^n$ checkerboard with any one square deleted, show by mathematical induction, that it is possible to cover the board with 3-square L-shaped pieces. For example, a 4×4 board with a corner deleted could be covered as follows,



Hint: The basis is easy — a 2×2 board with one square missing is covered by a single L-shaped piece. The trick is to see how an arbitrary 2^k board with one square missing can be constructed out of an L-shaped piece and 2^{k-1} size boards with a square missing. But this is not hard!

8.3 Further Examples (for Part III)

We continue our series of examples, moving now to quantificational cases, and to some theorems that will be useful especially if you go on to consider Part III.

8.3.1 Case

For variables x and v, where v does not appear in term t, it should be obvious that $[t^x_v]^v_x = t$. If we replace every instance of x with v, and then all the instances of v with x, we get back to where we started. The restriction that v not appear in t is required to prevent putting back instances of x where there were none in the original — as fxv^x_v is fvv, but then fvv^v_x is fxx. We demonstrate that when v does not appear in t, $[t^x_v]^v_x = t$ more rigorously by a simple induction on the number of function symbols in t. Suppose v does not appear in t.

(K) *Basis:* If t has no function symbols, then it is a variable or a constant. If it is a variable or a constant other than x, then $t^x_v = t$ (nothing is replaced); and since v does not appear in t, $t^v_x = t$ (nothing is replaced); so $[t^x_v]^v_x = t$. If t is the variable x, then $t^x_v = v$; and $v^v_x = x$; so $[t^x_v]^v_x = x = t$. In either case, then, $[t^x_v]^v_x = t$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, and v does not appear in t, then $[t^x_v]^v_x = t$.

Show: If t has k function symbols, and v does not appear in t, then $[t^x_v]^v_x = t$.

If t has k function symbols, then it is of the form, $h^n s_1 \dots s_n$ for some function symbol h^n and terms $s_1 \dots s_n$ each of which has $< k$ function symbols; since v does not appear in t, it does not appear in any of $s_1 \dots s_n$; so the inductive assumption applies to $s_1 \dots s_n$; so by assumption $[s_1{}^x_v]^v_x = s_1$, and ... and $[s_n{}^x_v]^v_x = s_n$. But $[t^x_v]^v_x = [h^n s_1 \dots s_n{}^x_v]^v_x$; and since replacements only occur within the terms, this is $h^n[s_1{}^x_v]^v_x \dots [s_n{}^x_v]^v_x$; and by assumption this is $h^n s_1 \dots s_n = t$. So $[t^x_v]^v_x = t$.

Indct: For any term t, if v does not appear in t, $[t^x_v]^v_x = t$

Consider a concrete application of the point that replacements occur only within the terms. We find $[f^2g^2axb\,{}^x_v]^v_x$ if we find $[g^2ax\,{}^x_v]^v_x$ and $[b^x_v]^v_x$ and compose the whole from them — for the function symbol f^2 cannot be affected by substitutions on the variables! It is also worthwhile to note the place where it matters that v is not a variable in t: In the basis case where t is a variable other than x, $t^v_x = t$ insofar as nothing is replaced; but suppose t is v; then $t^v_x = x \neq t$, and we do not achieve the desired result.

This result can be extended to one with application to formulas. If v is not free in a formula $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$. We require the restriction that v is not free in $\mathcal{P}$ for the same reason as before: if v were free in $\mathcal{P}$, we might end up with instances of x where there are none in the original — as Rxv^x_v is Rvv, but then Rvv^v_x is Rxx. And we need the restriction that v is free for x in $\mathcal{P}$ so that instances of x go back for all the instances of v when free instances of v are replaced by x — as $\forall vRxv^x_v$ is $\forall vRvv$, but then remains the same when x is substituted for free instances of v. Here is the basic structure of the argument, with parts left for homework.

T8.2. For variables x and v, if v is not free in a formula $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Let $\mathcal{P}$ be any formula such that v is not free $\mathcal{P}$ and v is free for x in $\mathcal{P}$. We show that $[\mathcal{P}^x_v]^v_x = \mathcal{P}$ by induction on the number of operator symbols in $\mathcal{P}$.

Basis: If $\mathcal{P}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \ldots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \ldots t_n$. (i) If $\mathcal{P}$ is $\mathcal{S}$ then it has no variables; so $\mathcal{S}^x_v = \mathcal{S}$ and $\mathcal{S}^v_x = \mathcal{S}$; so $[\mathcal{S}^x_v]^v_x = \mathcal{S}$; so $[\mathcal{P}^x_v]^v_x = \mathcal{P}$. (ii) If $\mathcal{P}$ is $\mathcal{R}^n t_1 \ldots t_n$ then $[\mathcal{P}^x_v]^v_x = \mathcal{R}^n[t_1\,{}^x_v]^v_x \ldots [t_n\,{}^x_v]^v_x$; but since v is not free in $\mathcal{P}$, v does not appear at all in $\mathcal{P}$ or its terms; so by the previous result, $[t_1\,{}^x_v]^v_x = t_1$ and $\ldots [t_n\,{}^x_v]^v_x = t_n$; so $[\mathcal{P}^x_v]^v_x = \mathcal{R}^n t_1 \ldots t_n$; which is to say, $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols, where v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Show: Any $\mathcal{P}$ with k operator symbols, is such that if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $(\mathcal{A} \to \mathcal{B})$ or $\forall w A$ for some variable w and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$, v is not free in $\mathcal{P}$, and v is free for x in $\mathcal{P}$. Then $[\mathcal{P}^x_v]^v_x = [(\sim\mathcal{A})^x_v]^v_x = \sim[\mathcal{A}^x_v]^v_x$. Since v is not free in $\mathcal{P}$, v is not free in $\mathcal{A}$; and since v is free for x in $\mathcal{P}$, v is free for x in $\mathcal{A}$. So the assumption applies to $\mathcal{A}$ ...

($\to$) Homework.

($\forall$) Suppose $\mathcal{P}$ is $\forall w\mathcal{A}$, v is not free in $\mathcal{P}$, and v is free for x in $\mathcal{P}$. Either x is free in $\mathcal{P}$ or not. (i) If x is not free in $\mathcal{P}$, then $\mathcal{P}^x_v = \mathcal{P}$ and since v is not free in $\mathcal{P}$, $\mathcal{P}^v_x = \mathcal{P}$; so $[\mathcal{P}^x_v]^v_x = \mathcal{P}$. (ii) Suppose x is free in $\mathcal{P} = \forall w\mathcal{A}$. Then x is other than w; and since v is free for x in $\mathcal{P}$, v is other than w; so the quantifier does not affect the replacements, and $[\mathcal{P}^x_v]^v_x$ is $\forall w[\mathcal{A}^x_v]^v_x$. Since v is not free in $\mathcal{P}$ and is not w, v is not free in $\mathcal{A}$; and since v is free for x in $\mathcal{P}$, v is free for x in $\mathcal{A}$. So the inductive assumption applies to $\mathcal{A}$; so $[\mathcal{A}^x_v]^v_x = \mathcal{A}$; so $\forall w[\mathcal{A}^x_v]^v_x = \forall w\mathcal{A}$; but this is just to say, $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

If $\mathcal{P}$ has k operator symbols, if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

Indct: For any $\mathcal{P}$, if v is not free in $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

There are a few things to note about this argument. First, again, we have to be careful that the formulas $\mathcal{A}$ and $\mathcal{B}$ of which $\mathcal{P}$ is composed are in fact of the sort to which the inductive assumption applies. In this case, the requirement is not only that $\mathcal{A}$ and $\mathcal{B}$ have $< k$ operator symbols, but that they satisfy the additional assumptions, that v is not free but is free for x. It is easy to see that this condition obtains in the cases for $\sim$ and $\to$, but it is relatively complicated in the case for $\forall$, where there is interaction with another quantifier. Observe also that we cannot assume that the arbitrary quantifier has the same variable as x or v. In fact, it is because the variable may be different that we are able to reason the way we do. Finally, observe that the arguments of this section for (K) and T8.2 are a "linked pair" in the sense that the result of the first for terms is required for the basis of the next for formulas. This pattern repeats in the next cases.

*E8.17. Provide a complete argument for T8.2, completing cases for ($\sim$) and ($\to$). You should set up the complete induction, but may appeal to the text at parts that are already completed, just as the text appeals to homework.

E8.18. Consider language $\mathcal{L}_t$ from examples (C) and (D) that has just function symbols f^2 and g^4, and let it be developed so that it has just one constant symbol c, and just the primitive operators $\mid$, from p. 333, and $\exists$. Provide a complete demonstration for expressions in this language that $[\mathcal{P}^x_v]^v_x = \mathcal{P}$. Hint: You will need arguments parallel to (K) and then T8.2, but structured by the symbols of this language.

8.3.2 Case

This example develops another pair of linked results which may seem obvious. Even so, the reasoning is instructive, and we will need the results for things to come. First,

T8.3. For any interpretation I, variable assignments d and h, and term t, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

If variable assignments agree at least on assignments to the variables in t, then corresponding term assignments agree on the assignment to t. The reasoning, as one might expect, is by induction on the number of function symbols in t. Let I, d, h and t be arbitrary, and suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t.

Basis: If t has no function symbols, then it is a variable x or a constant c. (i) Suppose t is a constant c. Then by TA(c), $\mathsf{I_d}[t] = \mathsf{I_d}[c] = \mathsf{I}[c]$; and by TA(c) again, $\mathsf{I}[c] = \mathsf{I_h}[c] = \mathsf{I_h}[t]$. So $\mathsf{I_d}[t] = \mathsf{I_h}[t]$. (ii) Suppose t is a variable x. Then by TA(v), $\mathsf{I_d}[t] = \mathsf{I_d}[x] = \mathsf{d}[x]$; but by the assumption to the theorem, $\mathsf{d}[x] = \mathsf{h}[x]$; and by TA(v) again, $\mathsf{h}[x] = \mathsf{I_h}[x] = \mathsf{I_h}[t]$. So $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, and $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

Show: If t has k function symbols, and $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

If t has k function symbols, then it is of the form $h^n s_1 \dots s_n$ for some function symbol h^n and terms $s_1 \dots s_n$ with $< k$ function symbols. Suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t. Since $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in $s_1 \dots s_n$; so the inductive assumption applies to $s_1 \dots s_n$. So $\mathsf{I_d}[s_1] = \mathsf{I_h}[s_1]$, and ... and $\mathsf{I_d}[s_n] = \mathsf{I_h}[s_n]$; so $\langle \mathsf{I_d}[s_1], \dots \mathsf{I_d}[s_n]\rangle = \langle \mathsf{I_h}[s_1], \dots \mathsf{I_h}[s_n]\rangle$; so $\mathsf{I}[h^n]\langle \mathsf{I_d}[s_1], \dots \mathsf{I_d}[s_n]\rangle = \mathsf{I}[h^n]$ $\langle \mathsf{I_h}[s_1], \dots \mathsf{I_h}[s_n]\rangle$; so by TA(f), $\mathsf{I_d}[h^n s_1 \dots s_n] = \mathsf{I_h}[h^n s_1 \dots s_n]$; which is to say $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

Indct: For any t, $\mathsf{I_d}[t] = \mathsf{I_h}[t]$.

So for any interpretation I, variable assignments d and h and term t, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable in t, then $\mathsf{I_d}[t] = \mathsf{I_h}[t]$. It should be clear that we follow our usual pattern to complete the show step: The assumption gives us information about the parts — in this case, about assignments to $s_1 \dots s_n$; from this, with TA, we move to a conclusion about the whole term t. Notice again, that it is important to show that the parts are of the right sort for the inductive assumption to apply: it matters that $s_1 \dots s_n$ have $< k$ function symbols, and that $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable in $s_1 \dots s_n$. Perhaps the overall result is intuitively obvious: If there is no difference in assignments to relevant variables, then there is no difference in assignments to the whole terms. Our proof merely makes explicit how this result follows from the definitions.

We now turn to a result that is very similar, except that it applies to formulas. In this case, T8.3 is essential for reasoning in the basis.

T8.4. For any interpretation I, variable assignments d and h, and formula $\mathcal{P}$, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

The argument, as you should expect, is by induction on the number of operator symbols in the formula $\mathcal{P}$. Let I, d, h and $\mathcal{P}$ be arbitrary, and suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x free in $\mathcal{P}$.

Basis: If $\mathcal{P}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \dots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$. (i) Suppose $\mathcal{P}$ is a sentence letter $\mathcal{S}$. Then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$; iff $\mathsf{I_d}[\mathcal{S}] = \mathsf{S}$; by SF(s) iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; by SF(s) again iff $\mathsf{I_h}[\mathcal{S}] = \mathsf{S}$; iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$. (ii) Suppose $\mathcal{P}$ is $\mathcal{R}^n t_1 \dots t_n$. Then since every variable in $\mathcal{P}$ is free, by the assumption for the theorem, $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable in $\mathcal{P}$; so $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable in $t_1 \dots t_n$; so by T8.3, $\mathsf{I_d}[t_1] = \mathsf{I_h}[t_1]$, and ... and $\mathsf{I_d}[t_n] = \mathsf{I_h}[t_n]$; so $\langle \mathsf{I_d}[t_1], \dots \mathsf{I_d}[t_n]\rangle = \langle \mathsf{I_h}[t_1], \dots \mathsf{I_h}[t_n]\rangle$; so $\langle \mathsf{I_d}[t_1], \dots \mathsf{I_d}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$ iff $\langle \mathsf{I_h}[t_1], \dots \mathsf{I_h}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; so by SF(r), $\mathsf{I_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$; which is to say, $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

Assp: For any i, $0 \le i < k$, if $\mathcal{P}$ has i operator symbols and $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

Show: If $\mathcal{P}$ has k operator symbols and $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \rightarrow \mathcal{B}$, or $\forall v \mathcal{A}$ for variable v and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols. Suppose $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$. Then since $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, and every variable free in $\mathcal{A}$ is free in $\mathcal{P}$, $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable in $\mathcal{A}$; so the inductive assumption applies to $\mathcal{A}$. $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\sim\mathcal{A}] = \mathsf{S}$; by SF($\sim$) iff $\mathsf{I_d}[\mathcal{A}] \neq \mathsf{S}$; by assumption iff $\mathsf{I_h}[\mathcal{A}] \neq \mathsf{S}$; by SF($\sim$), iff $\mathsf{I_h}[\sim\mathcal{A}] = \mathsf{S}$; iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

($\rightarrow$) Homework.

($\forall$) Suppose $\mathcal{P}$ is $\forall v \mathcal{A}$. Then since $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable in $\mathcal{A}$ with the possible exception of v; so for arbitrary $\mathsf{o} \in \mathsf{U}$, $\mathsf{d}(v|\mathsf{o})[x] = \mathsf{h}(v|\mathsf{o})[x]$ for every free variable x in $\mathcal{A}$. Since the assumption applies to arbitrary assignments, it applies to $\mathsf{d}(v|\mathsf{o})$ and $\mathsf{h}(v|\mathsf{o})$; so by assumption for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(v|\mathsf{o})}[\mathcal{A}] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{h}(v|\mathsf{o})}[\mathcal{A}] = \mathsf{S}$.

Now suppose $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ but $\mathsf{I_h}[\mathcal{P}] \neq \mathsf{S}$; then $\mathsf{I_d}[\forall v \mathcal{A}] = \mathsf{S}$ but $\mathsf{I_h}[\forall v \mathcal{A}] \neq \mathsf{S}$; from the latter, by SF($\forall$), there is some $\mathsf{o} \in \mathsf{U}$ such that $\mathsf{I}_{\mathsf{h}(v|\mathsf{o})}[\mathcal{A}] \neq \mathsf{S}$; let m be a particular individual of this sort; then $\mathsf{I}_{\mathsf{h}(v|\mathsf{m})}[\mathcal{A}] \neq \mathsf{S}$; so, as above, with the inductive assumption, $\mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathcal{A}] \neq \mathsf{S}$. But $\mathsf{I_d}[\forall v \mathcal{A}] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{I}_{\mathsf{d}(v|\mathsf{o})}[\mathcal{A}] = \mathsf{S}$; so $\mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathcal{A}] = \mathsf{S}$. This is impossible; reject the assumption: if $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$, then $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$. And similarly [by homework] in the other direction.

If $\mathcal{P}$ has k operator symbols, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

Indct: For any $\mathcal{P}$, $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$.

So for any interpretation I, variable assignments d and h, and formula $\mathcal{P}$, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_h}[\mathcal{P}] = \mathsf{S}$. Notice again that it is important to make sure the inductive assumption applies. In the ($\forall$) case, first we are careful to distinguish the arbitrary variable of quantification v, from x of the assumption. For the quantifier case, the condition that d and h agree on assignments to all the free variables in $\mathcal{A}$ is *not* satisfied merely because they agree on assignments to all the free variables in $\mathcal{P}$. We solve the problem by switching to assignments $\mathsf{d}(v|\mathsf{o})$ and $\mathsf{h}(v|\mathsf{o})$, which must agree on all the free variables in $\mathcal{A}$. (Why?) The overall reasoning in the quantifier case is fairly sophisticated. But you should be in a position to bear down and follow each step.

From T8.4 it is a short step to a corollary, the proof of which was promised in chapter 4: If a *sentence* $\mathscr{P}$ is satisfied on any variable assignment, then it is satisfied on every variable assignment, and so true.

T8.5. For any interpretation I and sentence $\mathscr{P}$, if there is some assignment d such that $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$, then $\mathsf{I}[\mathscr{P}] = \mathsf{T}$.

For sentence $\mathscr{P}$ and interpretation I, suppose there is some assignment d such that $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$, but $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$. From the latter, by TI, there is some particular assignment h such that $\mathsf{I_h}[\mathscr{P}] \neq \mathsf{S}$; but if $\mathscr{P}$ is a sentence, it has no free variables; so every assignment agrees with h in its assignment to every free variable in $\mathscr{P}$; in particular d agrees with h in its assignment to every free variable in $\mathscr{P}$; so by T8.4, $\mathsf{I_d}[\mathscr{P}] \neq \mathsf{S}$. This is impossible; reject the assumption: if $\mathsf{I_d}[\mathscr{P}] = \mathsf{S}$ then $\mathsf{I}[\mathscr{P}] = \mathsf{T}$.

In effect, the reasoning is as sketched in chapter 4. Whether $\forall x\mathscr{P}$ is satisfied by d does not depend on what particular object d assigns to x — for satisfaction of the quantified formula depends on satisfaction for *every* assignment to x. The key step is contained in the reasoning for the ($\forall$) case of the induction. Given this, the move to T8.5 is straightforward.

T8.5 puts us in a position to recover simple semantic conditions for *sentences* of the sort $\sim\mathscr{P}$ and $\mathscr{P} \rightarrow \mathscr{Q}$.

T8.6. For any sentences $\mathscr{P}$ and $\mathscr{Q}$, (i) $\mathsf{I}[\sim\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$; and (ii) $\mathsf{I}[\mathscr{P} \rightarrow \mathscr{Q}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathscr{Q}] = \mathsf{T}$.

($\sim$) Suppose $\mathsf{I}[\sim\mathscr{P}] = \mathsf{T}$; then by TI, for any d, $\mathsf{I_d}[\sim\mathscr{P}] = \mathsf{S}$; so by SF($\sim$), $\mathsf{I_d}[\mathscr{P}] \neq \mathsf{S}$; and by TI again, $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$. Suppose $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$; then by TI, there is some d such that $\mathsf{I_d}[\mathscr{P}] \neq \mathsf{S}$; let h be a particular assignment of this sort; then $\mathsf{I_h}[\mathscr{P}] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{I_h}[\sim\mathscr{P}] = \mathsf{S}$; and since $\mathscr{P}$ is a sentence, $\sim\mathscr{P}$ is a sentence; so by T8.5, $\mathsf{I}[\sim\mathscr{P}] = \mathsf{T}$. So $\mathsf{I}[\sim\mathscr{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$.

($\rightarrow$) Homework.

Thus for the sentential operators, sentences of a quantificational language obey the same semantic conditions as ones from sentential languages.

*E8.19. Provide a complete argument for T8.4, completing the case for ($\rightarrow$), and expanding the other direction for ($\forall$). You should set up the complete induction, but may appeal to the text at parts that are already completed, as the text appeals to homework.

E8.20. Complete the demonstration of T8.6 by working the case for ($\rightarrow$).

E8.21. Show that T8.4 holds for expressions in $\mathcal{L}_t$ from E8.18. Hint: you will need results parallel to both T8.3 and T8.4.

E8.22. Show that for any interpretation I and *sentence* $\mathcal{P}$, either $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ or $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$. Hint: This is not an argument by induction, but rather another corollary to T8.4. So begin by supposing the result is false....

8.3.3 Case

Finally, we turn to another pair of results, with reasoning like what we have already seen.

T8.7. For any formula $\mathcal{P}$, term t, constant c, and distinct variables v and x, $[\mathcal{P}^v_t]^c_x$ is the same formula as $[\mathcal{P}^c_x]^v_{t^c_x}$

Notice that switching t for v and then x for c is not the same as switching x for c and then t for v — for if t contains an instance of c, that instance of c is replaced in the first case, but not in the second. The proof breaks into two parts. (i) By induction on the number of function symbols in an arbitrary term r, we show that $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$. Given this, (ii) by induction on the number of operator symbols in an arbitrary formula $\mathcal{P}$, we show that $[\mathcal{P}^v_t]^c_x = [\mathcal{P}^c_x]^v_{t^c_x}$. Only part (i) is completed here; (ii) is left for homework. Suppose $v \neq x$.

Basis: If r has no function symbols, then it is either v, c or some other constant or variable.

(v) Suppose r is v. Then r^v_t is t and $[r^v_t]^c_x$ is t^c_x. But r^c_x is v; so $[r^c_x]^v_{t^c_x}$ is t^c_x. So $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

(c) Suppose r is c. Then r^v_t is c and $[r^v_t]^c_x$ is x. But r^c_x is x; and, since $v \neq x$, $[r^c_x]^v_{t^c_x}$ is x. So $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

(oth) Suppose r is some variable or constant other than v or c. Then $r^v_t = [r^v_t]^c_x = r$. Similarly, $r^c_x = [r^c_x]^v_{t^c_x} = r$. So $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

Assp: For any i, $0 \leq i < k$, if r has i function symbols, then $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

First Theorems of Chapter 8

T8.1 For any $\mathcal{P}$ whose operators are $\sim$, $\vee$, $\wedge$ and $\rightarrow$, $\mathcal{P}^*$ is in normal form and $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

T8.2 For variables x and v, if v is not free in a formula $\mathcal{P}$ and v is free for x in $\mathcal{P}$, then $[\mathcal{P}^x_v]^v_x = \mathcal{P}$.

T8.3 For any interpretation I, variable assignments d and h, and term t, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every variable x in t, then $\mathsf{I}_\mathsf{d}[t] = \mathsf{I}_\mathsf{h}[t]$.

T8.4 For any interpretation I, variable assignments d and h, and formula $\mathcal{P}$, if $\mathsf{d}[x] = \mathsf{h}[x]$ for every free variable x in $\mathcal{P}$, then $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I}_\mathsf{h}[\mathcal{P}] = \mathsf{S}$.

T8.5 For any interpretation I and sentence $\mathcal{P}$, if there is some assignment d such that $\mathsf{I}_\mathsf{d}[\mathcal{P}] = \mathsf{S}$, then $\mathsf{I}[\mathcal{P}] = \mathsf{T}$.

T8.6 For any sentences $\mathcal{P}$ and $\mathcal{Q}$, (i) $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$; and (ii) $\mathsf{I}[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathcal{Q}] = \mathsf{T}$.

T8.7 For any formula $\mathcal{P}$, term t, constant c, and distinct variables v and x, $[\mathcal{P}^v_t]^c_x$ is the same formula as $[\mathcal{P}^c_x]^v_{t^c_x}$

Show: If r has k function symbols, then $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

If r has k function symbols, then it is of the form, $h^n s_1 \ldots s_n$ for some function symbol h^n and terms $s_1 \ldots s_n$ each of which has $< k$ function symbols. In this case, $[r^v_t]^c_x = h^n([s_1{}^v_t]^c_x \ldots [s_n{}^v_t]^c_x)$. Similarly, $[r^c_x]^v_{t^c_x} = h^n([s_1{}^c_x]^v_{t^c_x} \ldots [s_n{}^c_x]^v_{t^c_x})$. But by assumption, $[s_1{}^v_t]^c_x = [s_1{}^c_x]^v_{t^c_x}$, and … and $[s_n{}^v_t]^c_x = [s_n{}^c_x]^v_{t^c_x}$; so $h^n([s_1{}^v_t]^c_x \ldots [s_n{}^v_t]^c_x) = h^n([s_1{}^c_x]^v_{t^c_x} \ldots [s_n{}^c_x]^v_{t^c_x})$; so $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

Indct: For any r, $[r^v_t]^c_x = [r^c_x]^v_{t^c_x}$.

You will find this result useful when you turn to the final proof of T8.7. That argument is a straightforward induction on the number of operator symbols in $\mathcal{P}$. For the case where $\mathcal{P}$ is of the form $\forall w \mathcal{A}$, notice that v is either w or it is not. On the one hand, if v is w, then $\mathcal{P} = \forall w A$ has no free instances of v so that $\mathcal{P}^v_t = \mathcal{P}$, and $[\mathcal{P}^v_t]^c_x = \mathcal{P}^c_x$; but, similarly, $\mathcal{P}^c_x$ has no free instances of v, so $[\mathcal{P}^c_x]^v_{t^c_x} = \mathcal{P}^c_x$. On the other hand, if v is a variable other than w, then $[\mathcal{P}^v_t]^c_x = \forall w [\mathcal{A}^v_t]^c_x$ and $[\mathcal{P}^c_x]^v_{t^c_x} = \forall w [\mathcal{A}^c_x]^v_{t^c_x}$ and you will be able to use the inductive assumption.

*E8.23. Complete the proof of T8.7 by showing by induction on the number of operator symbols in an arbitrary formula $\mathcal{P}$ that if v is distinct from x, then $[\mathcal{P}^v_t]^c_x = [\mathcal{P}^c_x]^v_{t^c_x}$.

E8.24. Show that T8.7 holds for expressions in $\mathcal{L}_t$ from E8.18.

E8.25. Set $\mathsf{U} = \{1\}$, $\mathsf{I}[\mathcal{S}] = \mathsf{T}$ for every sentence letter $\mathcal{S}$, $\mathsf{I}[\mathcal{R}^1] = \{1\}$ for every $\mathcal{R}^1$; $\mathsf{I}[\mathcal{R}^2] = \{\langle 1, 1\rangle\}$ for every $\mathcal{R}^2$; and in general, $\mathsf{I}[\mathcal{R}^n] = \mathsf{U}^n$ for every $\mathcal{R}^n$. Where $\mathcal{P}$ is any formula whose only operators are $\to$, $\wedge$, $\vee$, $\leftrightarrow$, $\forall$ and $\exists$, show by induction on the number of operators in $\mathcal{P}$ that $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$. Use this result to show that $\not\models \sim\mathcal{P}$. Hint: This is a quantificational version of E8.10.

E8.26. Where the only operator in formula $\mathcal{P}$ is $\leftrightarrow$, show that $\mathcal{P}$ is valid, $\models \mathcal{P}$ iff each atomic in $\mathcal{P}$ occurs an even number of times. For this, say formulas $\mathcal{P}$ and $\mathcal{Q}$ are *equivalent* just in case $\mathsf{I_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$. Then the argument breaks into three parts.

(i) Show *com*: $\mathcal{A} \leftrightarrow \mathcal{B}$ is equivalent to $\mathcal{B} \leftrightarrow \mathcal{A}$; *assoc* $\mathcal{A} \leftrightarrow (\mathcal{B} \leftrightarrow \mathcal{C})$ is equivalent to $(\mathcal{A} \leftrightarrow \mathcal{B}) \leftrightarrow \mathcal{C}$; and *sub* if $\mathcal{A}$ is equivalent to $\mathcal{B}$, then $\mathcal{B} \leftrightarrow \mathcal{C}$ is equivalent to $\mathcal{A} \leftrightarrow \mathcal{C}$. These are simple arguments in the style of chapter 7.

(ii) Suppose the only operator in formula $\mathcal{P}$ is $\leftrightarrow$, and $\mathcal{Q}$ and $\mathcal{R}$ are any formulas, whose only operator is $\leftrightarrow$, such that the atomics of $\mathcal{Q}$ plus the atomics of $\mathcal{R}$ are the same as the atomics of $\mathcal{P}$. Where $\mathcal{P}$ has at least one operator symbol, show by induction on the number of operator symbols in $\mathcal{P}$, that $\mathcal{P}$ is equivalent to $\mathcal{Q} \leftrightarrow \mathcal{R}$. Hint: If $\mathcal{P}$ is of the form $\mathcal{A} \leftrightarrow \mathcal{B}$, then you will be able to use the assumption to say that $\mathcal{A} \leftrightarrow \mathcal{B}$ is equivalent to some $(\mathcal{Q}_\mathcal{A} \leftrightarrow \mathcal{R}_\mathcal{A}) \leftrightarrow (\mathcal{Q}_\mathcal{B} \leftrightarrow \mathcal{R}_\mathcal{B})$ which sort the atomics of $\mathcal{A}$ and $\mathcal{B}$ into the atomics of $\mathcal{Q}$ and the atomics of $\mathcal{R}$. Then you can use (i) to force a form $(\mathcal{Q}_\mathcal{A} \leftrightarrow \mathcal{Q}_\mathcal{B}) \leftrightarrow (\mathcal{R}_\mathcal{A} \leftrightarrow \mathcal{R}_\mathcal{B})$. But you will also have to take account of (simplified) cases where $\mathcal{A}$ and $\mathcal{B}$ lack atomics from $\mathcal{Q}$ or from $\mathcal{R}$.

(iii) Where the only operator in formula $\mathcal{P}$ is $\leftrightarrow$, show by induction on the number of operators in $\mathcal{P}$, that $\mathcal{P}$ is valid, $\models \mathcal{P}$ iff each atomic in $\mathcal{P}$ occurs an even number of times. Hints: Say an atomic which occurs an odd number of times has an "unmatched" occurrence. Then, if $\mathcal{P}$ has k operator symbols, either (a) all of the atomics in $\mathcal{P}$ are matched, (b) $\mathcal{P}$ has both matched and unmatched atomics, or (c) $\mathcal{P}$ includes only unmatched atomics. In the first

two cases, you will be able to use result (ii) with the assumption. For (c) use (ii) to get an expression of the sort $\mathcal{A} \leftrightarrow \mathcal{B}$ where atomics in $\mathcal{A}$ are disjoint from the atomics in $\mathcal{B}$; then, depending on how you have set things up, you may not even need the inductive assumption.

E8.27. Show that any sentential form $\mathcal{P}$ whose only operators are $\sim$ and $\leftrightarrow$ and whose truth table has at least four rows, has an even number of Ts and Fs under its main operator. Hints: Reason by induction on the number of operators in $\mathcal{P}$ where $\mathcal{P}$ is (a subformula) on a table with at least four rows — so for atomics you may be sure that a table with at least four rows has an even number of Ts and Fs. The show step has cases for $\sim$ and $\leftrightarrow$. The former is easy, the latter is not.

8.4 Additional Examples (for Part IV)

Our primary motivation in this section is to practice doing mathematical induction. However, a final series of examples develop some results about Q that will be particularly useful if you go on to consider Part IV. As we have already mentioned (p. 307, compare E7.20), many true generalizations are not provable in Robinson Arithmetic. However, we shall be able to show that Q is generally adequate for some interesting classes of results. As you work through these results, you may find it convenient to refer to the final chapter 8 theorems reference on p. 421.

First we shall string together a series of results sufficient to show that Q correctly *decides* atomic sentences of $\mathcal{L}_{NT}$: where N is the standard interpretation for number theory and $\mathcal{P}$ is a sentence $s = t$, $s \leq t$ or $s < t$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathsf{Q} \vdash_{ND} \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathsf{Q} \vdash_{ND} \sim\mathcal{P}$. Observe that if $\mathcal{P}$ is atomic and a sentence, it has no variables.

8.4.1 Case

Let $\overline{\mathsf{n}}$ abbreviate $\overbrace{S \ldots S}^{\mathsf{n}\ S s}\emptyset$. So, for example, $\overline{2}$ is $SS\emptyset$, and $\overline{0}$ is just $\emptyset$. We begin with some simple results for the addition and multiplication of these numerals.

T8.8. For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a} + \mathsf{b} = \mathsf{c}$, then $\mathsf{Q} \vdash_{ND} \overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

By induction on the value of b. Recall that by Q3, $\mathsf{Q} \vdash_{ND} x + \emptyset = x$ and from Q4, $\mathsf{Q} \vdash_{ND} x + Sy = S(x + y)$. In addition, we depend on the general fact that, so long as $\mathsf{a} > 0$, $\mathsf{S}\overline{\mathsf{a} - 1}$ is the same numeral as $\overline{\mathsf{a}}$.

Basis: Suppose $\mathsf{b} = \mathsf{0}$ and $\mathsf{a} + \mathsf{b} = \mathsf{c}$; then $\mathsf{a} = \mathsf{c}$; but by Q3, Q $\vdash_{ND}$ $\overline{\mathsf{a}} + \emptyset = \overline{\mathsf{a}}$; so Q $\vdash_{ND}$ $\overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$ if $\mathsf{a} + \mathsf{i} = \mathsf{c}$, then Q $\vdash_{ND}$ $\overline{\mathsf{a}} + \overline{\mathsf{i}} = \overline{\mathsf{c}}$.

Show: If $\mathsf{a} + \mathsf{k} = \mathsf{c}$, then Q $\vdash_{ND}$ $\overline{\mathsf{a}} + \overline{\mathsf{k}} = \overline{\mathsf{c}}$.

Suppose $\mathsf{a} + \mathsf{k} = \mathsf{c}$. Since $\mathsf{k} > \mathsf{i}$, $\mathsf{k} > 0$. So $\overline{\mathsf{k}}$ is the same as $S\overline{\mathsf{k} - 1}$; and $\mathsf{a} + \mathsf{k} - 1 = \mathsf{c} - 1$; and by assumption Q $\vdash_{ND}$ $(\overline{\mathsf{a}} + \overline{\mathsf{k} - 1}) = \overline{\mathsf{c} - 1}$. By Q4, Q $\vdash_{ND}$ $(\overline{\mathsf{a}} + S\overline{\mathsf{k} - 1}) = S(\overline{\mathsf{a}} + \overline{\mathsf{k} - 1})$; but $S\overline{\mathsf{k} - 1}$ is $\overline{\mathsf{k}}$ so Q $\vdash_{ND}$ $(\overline{\mathsf{a}} + \overline{\mathsf{k}}) = S(\overline{\mathsf{a}} + \overline{\mathsf{k} - 1})$; so with =E, Q $\vdash_{ND}$ $(\overline{\mathsf{a}} + \overline{\mathsf{k}}) = S\overline{\mathsf{c} - 1} = \overline{\mathsf{c}}$. So Q $\vdash_{ND}$ $\overline{\mathsf{a}} + \overline{\mathsf{k}} = \overline{\mathsf{c}}$.

Indct: For any a, b and c, if $\mathsf{a} + \mathsf{b} = \mathsf{c}$, then Q $\vdash_{ND}$ $\overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$.

There are some manipulations to get the result, but the idea is simple: From the basis, $\overline{\mathsf{a}} + \emptyset = \overline{\mathsf{a}}$; then given the assumption for one value of b, we use Q4 to get the next. Observe that we informally manipulate objects in the universe by expressions of the sort, '$\mathsf{a} + \mathsf{b} = \mathsf{c}$' — but doing so is not itself to manipulate the corresponding expression of $\mathscr{L}_{NT}$ which would appear, '$\overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{c}}$'.

*T8.9. For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a} \times \mathsf{b} = \mathsf{c}$ then Q $\vdash_{ND}$ $\overline{\mathsf{a}} \times \overline{\mathsf{b}} = \overline{\mathsf{c}}$. By induction on the value of b.

Hint: You should come to as stage where you want to apply the assumption to $\overline{\mathsf{a}} \times \overline{\mathsf{k} - 1} + \overline{\mathsf{a}}$; but since $\mathsf{a} \times (\mathsf{k} - 1) = \mathsf{a} \times \mathsf{k} - \mathsf{a} = \mathsf{c} - \mathsf{a}$ the inductive assumption tells you that Q $\vdash_{ND}$ $\overline{\mathsf{a}} \times \overline{\mathsf{k} - 1} = \overline{\mathsf{c} - \mathsf{a}}$; and you will be able to apply T8.8 for the desired result.

*E8.28. Provide an argument to show T8.9.

8.4.2 Case

T8.10. For any $\mathsf{a}, \mathsf{b} \in \mathsf{U}$, if $\mathsf{a} \neq \mathsf{b}$, then Q $\vdash_{ND}$ $\overline{\mathsf{a}} \neq \overline{\mathsf{b}}$

Whenever $\mathsf{a} \neq \mathsf{b}$, there is some $\mathsf{d} > 0$ that is the *difference* between them. We show that for any n, Q $\vdash_{ND}$ $\overline{\mathsf{n}} \neq \overline{\mathsf{d} + \mathsf{n}}$. The the case when $\mathsf{n} = \mathsf{a}$ and $\mathsf{d} + \mathsf{n} = \mathsf{b}$ gives the desired result. Recall that according to Q1, Q $\vdash_{ND}$ $\sim(Sx = \overline{0})$; and from Q2, Q $\vdash_{ND}$ $(Sx = Sy) \rightarrow (x = y)$.

Suppose $\mathsf{a} \neq \mathsf{b}$; then $\mathsf{a} < \mathsf{b}$ or $\mathsf{b} < \mathsf{a}$; without loss of generality, suppose $\mathsf{a} < \mathsf{b}$; then there is some $\mathsf{d} > 0$ such that $\mathsf{d} + \mathsf{a} = \mathsf{b}$. By induction on n, we show Q $\vdash_{ND}$ $\overline{\mathsf{n}} \neq \overline{\mathsf{d} + \mathsf{n}}$; the case when $\mathsf{n} = \mathsf{a}$ gives Q $\vdash_{ND}$ $\overline{\mathsf{a}} \neq \overline{\mathsf{d} + \mathsf{a}}$; which is to say, Q $\vdash_{ND}$ $\overline{\mathsf{a}} \neq \overline{\mathsf{b}}$.

Basis: Suppose $\mathsf{n} = 0$. Since $\mathsf{d} > 0$, $\overline{\mathsf{d}} = S\overline{\mathsf{d}-1}$; and since $\mathsf{n} = 0$, $\overline{\mathsf{d}} = \overline{\mathsf{d}+\mathsf{n}}$. By Q1 with reflexivity, Q $\vdash_{ND}$ $\emptyset \neq S\overline{\mathsf{d}-1}$; so Q $\vdash_{ND}$ $\overline{\mathsf{n}} \neq \overline{\mathsf{d}} = \overline{\mathsf{d}+\mathsf{n}}$; so Q $\vdash_{ND}$ $\overline{\mathsf{n}} \neq \overline{\mathsf{d}+\mathsf{n}}$.

Assp: For $0 \leq \mathsf{i} < \mathsf{k}$, Q $\vdash_{ND}$ $\overline{\mathsf{i}} \neq \overline{\mathsf{d}+\mathsf{i}}$

Show: Q $\vdash_{ND}$ $\overline{\mathsf{k}} \neq \overline{\mathsf{d}+\mathsf{k}}$

In this case, both k and $\mathsf{d}+\mathsf{k}$ are > 0; so $\overline{\mathsf{k}}$ is $S\overline{\mathsf{k}-1}$ and $\overline{\mathsf{d}+\mathsf{k}}$ is $S\overline{\mathsf{d}+\mathsf{k}-1}$; by Q2, Q $\vdash_{ND}$ $S\overline{\mathsf{k}-1} = S\overline{\mathsf{d}+\mathsf{k}-1} \rightarrow \overline{\mathsf{k}-1} = \overline{\mathsf{d}+\mathsf{k}-1}$; but by assumption, Q $\vdash_{ND}$ $\overline{\mathsf{k}-1} \neq \overline{\mathsf{d}+\mathsf{k}-1}$; so by MT, Q $\vdash_{ND}$ $S\overline{\mathsf{k}-1} \neq S\overline{\mathsf{d}+\mathsf{k}-1}$; which is to say, Q $\vdash_{ND}$ $\overline{\mathsf{k}} \neq \overline{\mathsf{d}+\mathsf{k}}$.

Indct: For any n, Q $\vdash_{ND}$ $\overline{\mathsf{n}} \neq \overline{\mathsf{d}+\mathsf{n}}$.

So Q $\vdash_{ND}$ $\overline{\mathsf{a}} \neq \overline{\mathsf{d}+\mathsf{a}} = \overline{\mathsf{b}}$. In the basis, we show that Q proves the difference d between a and b is not equal to 0. Given this, at the show, Q proves that adding one to each side results in an inequality; and similarly adding one again results in an inequality until we get the result that Q proves that $\overline{\mathsf{a}} \neq \overline{\mathsf{b}}$. The demonstration that Q $\vdash_{ND}$ $\overline{\mathsf{a}} \neq \overline{\mathsf{b}}$ works so long as we start with d the difference between a and b.

The same basic strategy applies in a related case. But we need a preliminary theorem for one of the parts.

T8.11. Q $\vdash_{ND}$ $Sj + \overline{\mathsf{n}} = j + S\overline{\mathsf{n}}$.

Hint: this is a simple induction on n. You will want the assumption in the form, Q $\vdash_{ND}$ $Sj + \overline{\mathsf{k}-1} = j + S\overline{\mathsf{k}-1} = j + \overline{\mathsf{k}}$.

Now we are ready for the result like T8.10.

T8.12. (i) If $\mathsf{a} \not\leq \mathsf{b}$, then Q $\vdash_{ND}$ $\overline{\mathsf{a}} \not\leq \overline{\mathsf{b}}$; and (ii) If $\mathsf{a} \not< \mathsf{b}$, then Q $\vdash_{ND}$ $\overline{\mathsf{a}} \not< \overline{\mathsf{b}}$.

Recall that $s \leq t$ is $\exists v(v + s = t)$ and $s < t$ is $\exists v(Sv + s = t)$ for v not in s or t. Suppose $\mathsf{a} \not\leq \mathsf{b}$ then $\mathsf{a} > \mathsf{b}$; so, again, there is a difference d between them.

For (i) we need that if $\mathsf{a} \not\leq \mathsf{b}$ then Q $\vdash_{ND}$ $\sim\exists v(v + \overline{\mathsf{a}} = \overline{\mathsf{b}})$. Suppose $\mathsf{a} \not\leq \mathsf{b}$; then $\mathsf{a} > \mathsf{b}$; so for $\mathsf{d} > 0$, $\mathsf{a} = \mathsf{d}+\mathsf{b}$. By induction on n, we show that for any n, Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$; the case when $\mathsf{n} = \mathsf{b}$ gives Q $\vdash_{ND}$ $j + \overline{\mathsf{a}} \neq \overline{\mathsf{b}}$; then by $\forall$I, Q $\vdash_{ND}$ $\forall v(v + \overline{\mathsf{a}} \neq \overline{\mathsf{b}})$; and the result follows by QN.

Basis: Suppose $\mathsf{n} = 0$; then $\overline{\mathsf{d}+\mathsf{n}} = \overline{\mathsf{d}}$; since $\mathsf{d} > 0$, $\overline{\mathsf{d}} = S\overline{\mathsf{d}-1}$. By Q1, Q $\vdash_{ND}$ $S(j + \overline{\mathsf{d}-\mathsf{n}}) \neq \emptyset$; but by Q4, Q $\vdash_{ND}$ $j + S\overline{\mathsf{d}-1} = S(j + \overline{\mathsf{d}-1})$; so Q $\vdash_{ND}$

$j + S\overline{\mathsf{d}-1} \neq \emptyset$; but this is just to say Q $\vdash_{ND}$ $j + \overline{\mathsf{d}} = j + \overline{\mathsf{d}+\mathsf{n}} \neq \emptyset = \overline{\mathsf{n}}$; so Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$.

Assp: For $0 \leq \mathsf{i} < \mathsf{k}$, Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{i}} \neq \overline{\mathsf{i}}$.

Show: Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{k}} \neq \overline{\mathsf{k}}$.

In this case, k and $\mathsf{d}+\mathsf{k} > 0$ so that $\overline{\mathsf{k}} = S\overline{\mathsf{k}-1}$ and $\overline{\mathsf{d}+\mathsf{k}} = S\overline{\mathsf{d}+\mathsf{k}-1}$. By assumption, Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{k}-1} \neq \overline{\mathsf{k}-1}$. But by Q2, Q $\vdash_{ND}$ $S(j + \overline{\mathsf{d}+\mathsf{k}-1}) = S\overline{\mathsf{k}-1} \to j + \overline{\mathsf{d}+\mathsf{k}-1} = \overline{\mathsf{k}-1}$; so by MT, Q $\vdash_{ND}$ $S(j + \overline{\mathsf{d}+\mathsf{k}-1}) \neq S\overline{\mathsf{k}-1}$; by Q4, Q $\vdash_{ND}$ $j + S\overline{\mathsf{d}+\mathsf{k}-1} = S(j + \overline{\mathsf{d}+\mathsf{k}-1})$; so Q $\vdash_{ND}$ $j + S\overline{\mathsf{d}+\mathsf{k}-1} \neq S\overline{\mathsf{k}-1}$; but this is just to say, Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{k}} \neq \overline{\mathsf{k}}$.

Indct: For any n, Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$

So Q $\vdash_{ND}$ $j + \overline{\mathsf{d}+\mathsf{b}} \neq \overline{\mathsf{b}}$ which is to say Q $\vdash_{ND}$ $j + \overline{\mathsf{a}} \neq \overline{\mathsf{b}}$. So by $\forall$I, Q $\vdash_{ND}$ $\forall v(v + \overline{\mathsf{a}} \neq \overline{\mathsf{b}})$; and by QN, Q $\vdash_{ND}$ $\sim\exists v(v + \overline{\mathsf{a}} = \overline{\mathsf{b}})$; which is to say, Q $\vdash_{ND}$ $\overline{\mathsf{a}} \not\leq \overline{\mathsf{b}}$. In the basis, we show that for $\mathsf{d} > 0$, Q proves $j + \overline{\mathsf{d}} \neq 0$. Then, at the show, each side is incremented by one until Q proves $j + \overline{\mathsf{a}} \neq \overline{\mathsf{b}}$. Again, this works because we begin with d the difference between a and b.

E8.29. Provide arguments to show T8.11 and then (ii) of T8.12. Hint: For the latter, the induction is to show Q $\vdash_{ND}$ $Sj + \overline{\mathsf{d}+\mathsf{n}} \neq \overline{\mathsf{n}}$. There is a complication, however, in the basis. From $\mathsf{a} \not< \mathsf{b}$, $\mathsf{a} = \mathsf{b}+\mathsf{d}$ for $\mathsf{d} \geq 0$. So we cannot set $\overline{\mathsf{d}} = S\overline{\mathsf{d}-1}$. You can solve the problem by obtaining T8.11 as a preliminary result. Then it will be easy to show $j + S\overline{\mathsf{d}} \neq \emptyset$ and apply the preliminary theorem. For the show, since $\mathsf{k} > 0$, the argument remains straightforward.

8.4.3 Case

Up to this stage, we have been dealing entirely with atomics whose only terms are numerals of the sort $\overline{\mathsf{n}}$. We now broaden our results to include atomic sentences with arbitrary terms.

We have said a formula is *true* iff it is satisfied on every variable assignment. Let us introduce a parallel notion for terms.

AI The *assignment* of a term on an interpretation $\mathsf{I}[t] = \mathsf{n}$ iff with any d for I, $\mathsf{I}_{\mathsf{d}}[t] = \mathsf{n}$.

In particular, from T8.3, if assignments d and h agree on assignments to free variables in t, then $\mathsf{I}_{\mathsf{d}}[t] = \mathsf{I}_{\mathsf{h}}[t]$; so if t is without free variables, any assignments must agree

on assignments to all the free variables in t. So it is automatic that, for a variable free term, any $\mathsf{I_d}[t] = \mathsf{I_h}[t] = \mathsf{I}[t]$.

Given this, we start by establishing that Q proves the proper relation between arbitrary variable free terms and numerals.

T8.13. For any variable-free term t of $\mathscr{L}_{NT}$, if $\mathsf{N}[t] = \mathsf{n}$, then $Q \vdash_{ND} t = \overline{\mathsf{n}}$

By induction on the number of function symbols in t.

Basis: If a variable-free term t has no function symbols, then it is the constant $\emptyset$. $\mathsf{N}[\emptyset] = \mathsf{0}$. But by =I, $Q \vdash_{ND} \emptyset = \emptyset$; so $Q \vdash_{ND} t = \overline{\mathsf{n}}$.

Assp: For any $i, 0 \leq i < k$ if t has i function symbols and $\mathsf{N}[t] = \mathsf{n}$, then $Q \vdash_{ND} t = \overline{\mathsf{n}}$.

Show: If t has k function symbols and $\mathsf{N}[t] = \mathsf{n}$, then $Q \vdash_{ND} t = \overline{\mathsf{n}}$.

If t has k function symbols, it is of the form, $S r$, $r + s$ or $r \times s$ for r, s with < k function symbols.

(S) t is $S r$. Suppose $\mathsf{N}[t] = \mathsf{n}$. Since r is variable free, $\mathsf{N}[r] = \mathsf{N_d}[r] = \mathsf{a}$ for some a. Since t is variable-free, $\mathsf{N}[t] = \mathsf{N_d}[t] = \mathsf{N_d}[S r]$; by TA(f), $\mathsf{N_d}[S r] = \mathsf{N}[S]\langle \mathsf{a} \rangle = \mathsf{a} + 1$; so $\mathsf{N}[t] = \mathsf{a} + 1$; so $\mathsf{a} + 1 = \mathsf{n}$. By assumption $Q \vdash_{ND} r = \overline{\mathsf{a}}$; but $Q \vdash_{ND} S r = S r$; so by =E, $Q \vdash_{ND} S r = S\overline{\mathsf{a}} = \overline{\mathsf{a} + 1} = \overline{\mathsf{n}}$; so $Q \vdash_{ND} t = \overline{\mathsf{n}}$.

(+) t is $r + s$. Suppose $\mathsf{N}[t] = \mathsf{n}$. Since r and s are variale-free, $\mathsf{N}[r] = \mathsf{N_d}[r] = \mathsf{a}$ and $\mathsf{N}[s] = \mathsf{N_d}[s] = \mathsf{b}$ for some a and b. Since t is variable free, $\mathsf{N}[t] = \mathsf{N_d}[t] = \mathsf{N_d}[r + s]$; by TA(f), $\mathsf{N_d}[r + s] = \mathsf{N}[+]\langle \mathsf{a}, \mathsf{b} \rangle = \mathsf{a} + \mathsf{b}$; so $\mathsf{N}[t] = \mathsf{a} + \mathsf{b}$; so $\mathsf{a} + \mathsf{b} = \mathsf{n}$. By assumption, $Q \vdash_{ND} r = \overline{\mathsf{a}}$ and $Q \vdash_{ND} s = \overline{\mathsf{b}}$; but by =I, $Q \vdash_{ND} r + s = r + s$; so by =E, $Q \vdash_{ND} r + s = \overline{\mathsf{a}} + \overline{\mathsf{b}}$; and since $\mathsf{a} + \mathsf{b} = \mathsf{n}$ by T8.8, $Q \vdash_{ND} \overline{\mathsf{a}} + \overline{\mathsf{b}} = \overline{\mathsf{n}}$; so $Q \vdash_{ND} r + s = \overline{\mathsf{n}}$. So $Q \vdash_{ND} t = \overline{\mathsf{n}}$.

($\times$) Similarly by homework.

Indct: So for any variable-free term t, with $\mathsf{N}[t] = \mathsf{n}$, $Q \vdash_{ND} t = \overline{\mathsf{n}}$

Our intended result, that Q correctly decides atomic sentences of $\mathscr{L}_{NT}$ is not an argument by induction, but rather collects what we have done into a simple argument.

T8.14. Q correctly decides atomic sentences of $\mathscr{L}_{NT}$. For any sentence $\mathscr{P}$ of the sort $s = t$, $s \leq t$ or $s < t$, if $\mathsf{N}[\mathscr{P}] = \mathsf{T}$ then $Q \vdash_{ND} \mathscr{P}$; and if $\mathsf{N}[\mathscr{P}] \neq \mathsf{T}$ then $Q \vdash_{ND} \sim\mathscr{P}$.

Since the atomics are sentences (and the quantified variable does not appear in the terms for the inequalities), s and t are variable free. A few selected parts are worked as examples.

(a) $\mathsf{N}[s = t] = \mathsf{T}$. Then by TI, for any d, $\mathsf{N_d}[s = t] = \mathsf{S}$; so by SF(r), $\langle \mathsf{N_d}[s], \mathsf{N_d}[t] \rangle \in \mathsf{N}[=]$; so $\mathsf{N_d}[s] = \mathsf{N_d}[t]$. But since s and t are variable free, for some a, $\mathsf{N}[s] = \mathsf{N_d}[s] = \mathsf{a} = \mathsf{N_d}[t] = \mathsf{N}[t]$; so by T8.13, $Q \vdash_{ND} s = \overline{\mathsf{a}}$ and $Q \vdash_{ND} t = \overline{\mathsf{a}}$; but by =I, $Q \vdash_{ND} \overline{\mathsf{a}} = \overline{\mathsf{a}}$ so by =E, $Q \vdash_{ND} s = t$.

(b) $\mathsf{N}[s = t] \neq \mathsf{T}$.

(c) $\mathsf{N}[s \leq t] = \mathsf{T}$. Then $\mathsf{N}[\exists v(v + s = t)] = \mathsf{T}$; so by TI, for any d, $\mathsf{N_d}[\exists v(v + s = t)] = \mathsf{S}$; so by SF($\exists$), for some $\mathsf{m} \in \mathsf{U}$, $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[v + s = t] = \mathsf{S}$; but $\mathsf{d}(v|\mathsf{m})[v] = \mathsf{m}$; and by TA(v), $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[v] = \mathsf{m}$; and since s and t are variable-free, $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[s] = \mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[t] = \mathsf{N}[t] = \mathsf{b}$ for some a and b. By TA(f), $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[v + s] = \mathsf{N}[+]\langle \mathsf{m}, \mathsf{a} \rangle = \mathsf{m} + \mathsf{a}$; and by SF(r), $\langle \mathsf{m}+\mathsf{a}, \mathsf{b} \rangle \in \mathsf{N}[=]$; so $\mathsf{m}+\mathsf{a} = \mathsf{b}$. From the latter, by T8.8, $Q \vdash_{ND} \overline{\mathsf{m}}+\overline{\mathsf{a}} = \overline{\mathsf{b}}$. So by $\exists$I, $Q \vdash_{ND} \exists v(v + \overline{\mathsf{a}} = \overline{\mathsf{b}})$; which is to say, $Q \vdash_{ND} \overline{\mathsf{a}} \leq \overline{\mathsf{b}}$. But since $\mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}[t] = \mathsf{b}$, by T8.13, $Q \vdash_{ND} s = \overline{\mathsf{a}}$ and $Q \vdash_{ND} t = \overline{\mathsf{b}}$; so by =E, $Q \vdash_{ND} s \leq t$.

(d) $\mathsf{N}[s \leq t] \neq \mathsf{T}$. Then $\mathsf{N}[\exists v(v + s = t)] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{N_d}[\exists v(v + s = t)] \neq \mathsf{S}$; so by SF($\exists$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{N}_{\mathsf{d}(v|\mathsf{o})}[v + s = t] \neq \mathsf{S}$; let m be an arbitrary individual of this sort; then $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[v + s = t] \neq \mathsf{S}$. $\mathsf{d}(v|\mathsf{m})[v] = \mathsf{m}$; so by TA(v), $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[v] = \mathsf{m}$; and since s and t are variable-free, $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[s] = \mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[t] = \mathsf{N}[t] = \mathsf{N}[t] = \mathsf{b}$ for some a and b. By TA(f), $\mathsf{N}_{\mathsf{d}(v|\mathsf{m})}[v + s] = \mathsf{N}[+]\langle \mathsf{m}, \mathsf{a} \rangle = \mathsf{m} + \mathsf{a}$; so that by SF(r), $\langle \mathsf{m} + \mathsf{a}, \mathsf{b} \rangle \notin \mathsf{N}[=]$; so $\mathsf{m} + \mathsf{a} \neq \mathsf{b}$; and since m is arbitrary, for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{o} + \mathsf{a} \neq \mathsf{b}$; so $\mathsf{a} \not\leq \mathsf{b}$; so by T8.12, $Q \vdash_{ND} \overline{\mathsf{a}} \not\leq \overline{\mathsf{b}}$. But since $\mathsf{N}[s] = \mathsf{a}$ and $\mathsf{N}[t] = \mathsf{b}$, by T8.13, $Q \vdash_{ND} s = \overline{\mathsf{a}}$ and $Q \vdash_{ND} t = \overline{\mathsf{b}}$; so by =E, $Q \vdash_{ND} s \not\leq t$.

(e) $\mathsf{N}[s < t] = \mathsf{T}$.

(f) $\mathsf{N}[s < t] \neq \mathsf{T}$.

Since we are able to correctly decide the required results at the level of numerals, and then equalities between numerals and arbitrary terms, we are able to combine the two to correctly decide arbitrary atomics.

E8.30. Complete the argument for T8.13 by completing the case for ($\times$). You should set up the entire induction, but may appeal to the text for parts that are already completed, just as the text appeals to homework.

E8.31. Complete the remaining cases of T8.14 to show that Q correctly decides atomic sentences of $\mathcal{L}_{NT}$.

8.4.4 Case

We conclude the chapter with some more examples of mathematical induction, this time working toward important results about inequality. We begin by aiming at a result sometimes called *trichotomy*, for any n, $\mathsf{Q} \vdash_{ND} \forall x(x < \overline{\mathsf{n}} \vee x = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < x)$. Again, though, we begin with preliminaries. Recall that the *bounded* quantifiers $(\forall x < t)\mathcal{P}$, $(\exists x < t)\mathcal{P}$, $(\forall x \leq t)\mathcal{P}$, and $(\exists x \leq t)\mathcal{P}$, are abbreviations with associated derived introduction and exploitation rules (see p. 301). First, a simple argument that repeats a pattern of reasoning we shall see again.

T8.15. For any n and T, if $T \vdash_{ND} x = Sy$ and $T \vdash_{ND} y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{n}}$, then $T \vdash_{ND} x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{n}}$.

The argument is by induction on the value of n. Suppose $T \vdash_{ND} x = Sy$.

Basis: $\mathsf{n} = 0$. Suppose $T \vdash_{ND} y = \overline{0}$; we need that $T \vdash_{ND} x = S\overline{0}$. But this is immediate by =E.

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$, if $T \vdash_{ND} y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{i}}$, then $T \vdash_{ND} x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{i}}$

Show: If $T \vdash_{ND} y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{k}}$, then $T \vdash_{ND} x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{k}}$. Suppose $T \vdash_{ND} y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{k}}$.

1.	$x = Sy$	given from T
2.	$y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{k}-1} \vee y = \overline{\mathsf{k}}$	given from T
3.	$y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{k}-1}$	A (g 2$\vee$E)
4.	$x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{k}-1}$	1,3 assp
5.	$x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{k}-1} \vee x = S\overline{\mathsf{k}}$	4 $\vee$I
6.	$y = \overline{\mathsf{k}}$	A (g 2$\vee$E)
7.	$x = S\overline{\mathsf{k}}$	1,6 =E
8.	$x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{k}}$	7 $\vee$I
9.	$x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{k}}$	2,3-5,6-8 $\vee$E

So $T \vdash_{ND} x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{k}}$.

Indct: For any n, if $T \vdash_{ND} x = Sy$ and $T \vdash_{ND} y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{\mathsf{n}}$, then $T \vdash_{ND} x = S\overline{0} \vee x = S\overline{1} \vee \ldots \vee x = S\overline{\mathsf{n}}$.

Intuitively, we can use $x = Sy$ together with an extended version of $\vee$E on $y = \overline{0} \vee y = \overline{1} \vee \ldots \vee y = \overline{n}$ to get the result. The induction works by obtaining the result for the first disjunct, and then showing that no matter how far we have gone, it is always possible to go to the next stage. This theorem is useful for the next.

T8.16. For any n, (i) $Q \vdash_{ND} (\forall x \leq \overline{n})(x = \overline{0} \vee x = \overline{1} \ldots \vee x = \overline{n})$ and (ii) $Q \vdash_{ND} (\forall x < \overline{n})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \ldots \vee x = \overline{n-1})$.

The first disjunct $\emptyset \neq \emptyset$ in (ii) is to guarantee that the result is a well-formed sentence, even when $n = 0$. We work part (ii). By induction on n.

Basis: We need to show $(\forall x < \emptyset)(\emptyset \neq \emptyset)$. But this is easy with T6.47.

1.	$j < \emptyset$	A (g $(\forall$I))
2.	$\emptyset = \emptyset$	A (c $\sim$I)
3.	$j \not< \emptyset$	from T6.47
4.	$\perp$	1,3 $\perp$I
5.	$\emptyset \neq \emptyset$	2-4 $\sim$I
6.	$(\forall x < \emptyset)(\emptyset \neq \emptyset)$	1-5 ($\forall$I)

Assp: For $0 \leq i < k$, $Q \vdash_{ND} (\forall x < \overline{i})(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{i-1})$

Show: $Q \vdash_{ND} (\forall x < \overline{k})(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{k-1})$. When $i = k - 1$ by assumption $Q \vdash_{ND} \emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{k-1-1})$; observe that in the case when $i = 0$ ($k = 1$) this series remains defined but reduces to $\emptyset \neq \emptyset$ since it contains all the members "up" to $k-1-1$ and there are not any; when $i = 1$ ($k = 2$) the series is $\emptyset \neq \emptyset \vee x = \overline{0}$; and so forth. Here are the main outlines of the derivation.

1.	$(\forall x < \overline{\mathsf{k}-1})(\emptyset \neq \emptyset \vee x = \overline{0} \vee \ldots \vee x = \overline{\mathsf{k}-1-1})$	by assp
2.	$j < \overline{\mathsf{k}}$	A (g, →I)
3.	$j = \overline{0} \vee \exists y(j = Sy)$	from Q7
4.	$j = \overline{0}$	A (g 3∨E)
5.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{k}-1}$	4 ∨I
6.	$\exists y(j = Sy)$	A (g 3∨E)
7.	$j = Sl$	A (g 6∃E)
8.	$\exists v(Sv + j = \overline{\mathsf{k}})$	2 abv
9.	$Sh + j = \overline{\mathsf{k}}$	A (g 8∃E)
10.	$Sh + Sl = \overline{\mathsf{k}}$	7,9 =E
11.	$S(Sh + l) = S\overline{\mathsf{k}-1}$	10 with Q4
12.	$Sh + l = \overline{\mathsf{k}-1}$	11 with Q2
13.	$\exists v(Sv + l = \overline{\mathsf{k}-1})$	12 ∃I
14.	$l < \overline{\mathsf{k}-1}$	13 abv
15.	$l < \overline{\mathsf{k}-1}$	8,9-14∃E
16.	$\emptyset \neq \emptyset \vee l = \overline{0} \vee \ldots \vee l = \overline{\mathsf{k}-1-1}$	1,15 (∀E)
17.	$\emptyset \neq \emptyset \vee j = \overline{1} \vee \ldots \vee j = k-1$	7,16 with T8.15
18.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee j = \overline{1} \vee \ldots \vee j = \overline{\mathsf{k}-1}$	17, ∨I
19.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee j = \overline{1} \vee \ldots \vee j = \overline{\mathsf{k}-1}$	6,7-18 ∃E
20.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee j = \overline{1} \vee \ldots \vee j = \overline{\mathsf{k}-1}$	3,4-5,6-19 ∨E
21.	$(\forall x < \overline{\mathsf{k}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{k}-1})$	2-20 (∀I)

So Q $\vdash_{ND}$ $(\forall x < \overline{\mathsf{k}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{\mathsf{k}-1})$.

Indct: So for any n, Q $\vdash_{ND}$ $(\forall x < \overline{\mathsf{n}})(\emptyset \neq \emptyset \vee x = \overline{0} \vee x = \overline{1} \ldots \vee x = \overline{\mathsf{n}-1})$

From Q7, either j is zero or it is not. If j is zero, then the result is easy. If j is a successor, then (with a little work), there is an $l < \overline{\mathsf{k}-1}$ to which we may apply the assumption; once we have done that, it is a short step to the result again.

E8.32. Complete the demonstration of T8.16 by showing part (i). Hint: You have the basis already from T6.46.

8.4.5 Case

The next theorem is a sort of mirror to T8.16, and illustrates a pattern of reasoning we have already seen in application to extended disjunctions.

T8.17. For any n, (i) $\mathsf{Q} \vdash_{ND} \forall x[(x = \overline{\mathsf{0}} \vee x = \overline{\mathsf{1}} \ldots \vee x = \overline{\mathsf{n}}) \rightarrow x \leq \overline{\mathsf{n}}]$ and (ii) $\mathsf{Q} \vdash_{ND} \forall x[(\emptyset \neq \emptyset \vee x = \overline{\mathsf{0}} \vee \ldots \vee x = \overline{\mathsf{n}-\mathsf{1}}) \rightarrow x < \overline{\mathsf{n}}]$

Again I illustrate just (ii). For any n and $\mathsf{a} \leq \mathsf{n}$ we show by induction on the value of a that $\mathsf{Q} \vdash_{ND} (\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{a}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{n}}$; the case when $\mathsf{a} = \mathsf{n}$ gives $\mathsf{Q} \vdash_{ND} (\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{n}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{n}}$; and the desired result follows immediately by $\forall$I. Observe that a when $\mathsf{a} = \mathsf{0}$ the series reduces to $\emptyset \neq \emptyset$ as before.

Basis: $\mathsf{a} = \mathsf{0}$. We need $\mathsf{Q} \vdash_{ND} \emptyset \neq \emptyset \rightarrow j < \overline{\mathsf{n}}$

1.	$\emptyset \neq \emptyset$	A (1 $\rightarrow$I)
2.	$j \not< \overline{\mathsf{n}}$	A (c $\sim$E)
3.	$\emptyset = \emptyset$	=I
4.	$\perp$	3,1 $\perp$I
5.	$j < \overline{\mathsf{n}}$	2-4 $\sim$E
6.	$\emptyset \neq \emptyset \rightarrow j < \overline{\mathsf{n}}$	1-5 $\rightarrow$I

Assp: For any i, $\mathsf{0} \leq \mathsf{i} < \mathsf{k} \leq \mathsf{n}$, $\mathsf{Q} \vdash_{ND} (\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{i}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{n}}$

Show: $\mathsf{Q} \vdash_{ND} (\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{k}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{n}}$

1.	$(\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{k}-\mathsf{1}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{n}}$	assp
2.	$\emptyset \neq \emptyset \vee j = \emptyset \vee \ldots \vee j = \overline{\mathsf{k}-\mathsf{1}-\mathsf{1}} \vee j = \overline{\mathsf{k}-\mathsf{1}}$	A (g $\rightarrow$I)
3.	$\emptyset \neq \emptyset \vee j = \emptyset \vee \ldots \vee j = \overline{\mathsf{k}-\mathsf{1}-\mathsf{1}}$	A (g 2$\vee$E)
4.	$j < \overline{\mathsf{n}}$	1,3 $\rightarrow$E
5.	$j = \overline{\mathsf{k}-\mathsf{1}}$	A (g, 1$\vee$E)
6.	$\overline{\mathsf{k}-\mathsf{1}} < \overline{\mathsf{n}}$	T8.14 ($\mathsf{k} \leq \mathsf{n}$)
7.	$j < \overline{\mathsf{n}}$	6,5 =E
8.	$j < \overline{\mathsf{n}}$	2,3-4,5-7 $\vee$E
9.	$(\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{k}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{n}}$	2-8 $\rightarrow$I

So $\mathsf{Q} \vdash_{ND} (\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{k}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{k}}$.

Indct: For any n, $\mathsf{Q} \vdash_{ND} (\emptyset \neq \emptyset \vee j = \overline{\mathsf{0}} \vee \ldots \vee j = \overline{\mathsf{n}-\mathsf{1}}) \rightarrow j < \overline{\mathsf{n}}$.

So by $\forall$I, $\mathsf{Q} \vdash_{ND} \forall x[(\emptyset \neq \emptyset \vee x = \overline{\mathsf{0}} \vee \ldots \vee x = \overline{\mathsf{n}-\mathsf{1}}) \rightarrow x < \overline{\mathsf{n}}]$. The basis is easy. Once we set it up by $\vee$E, the show is easy too. Observe the use of T8.14 in the second case: since $\mathsf{k} \leq \mathsf{n}$, $\mathsf{k} - 1 < \mathsf{n}$; so by T8.14, $\mathsf{Q} \vdash_{ND} \overline{\mathsf{k}-\mathsf{1}} < \overline{\mathsf{n}}$. The next theorem does not require mathematical induction at all.

T8.18. For any n, (i) Q $\vdash_{ND}$ $\forall x[\overline{\mathsf{n}} \leq x \rightarrow (\overline{\mathsf{n}} = x \vee S\overline{\mathsf{n}} \leq x)]$ and (ii) $\forall x[\overline{\mathsf{n}} < x \rightarrow (S\overline{\mathsf{n}} = x \vee S\overline{\mathsf{n}} < x)]$.

Again I illustrate (ii).

1.	$\overline{\mathsf{n}} < j$	A (g →I)
2.	$\exists v(Sv + \overline{\mathsf{n}} = j)$	1 abv
3.	$Sk + \overline{\mathsf{n}} = j$	A (g 2∃E)
4.	$k = \emptyset \vee \exists y(k = Sy)$	from Q7
5.	$k = \emptyset$	A (g 4∨E)
6.	$S\emptyset + \overline{\mathsf{n}} = j$	3,5 =E
7.	$S\emptyset + \overline{\mathsf{n}} = S\overline{\mathsf{n}}$	from T8.8
8.	$j = S\overline{\mathsf{n}}$	6,7 =E
9.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < \overline{\mathsf{j}}$	8 ∨I
10.	$\exists y(k = Sy)$	A (g 4∨E)
11.	$k = Sl$	A (g 10∃E)
12.	$k + S\overline{\mathsf{n}} = j$	from 3 with T8.11
13.	$Sl + S\overline{\mathsf{n}} = j$	12,11 =E
14.	$\exists v(Sv + S\overline{\mathsf{n}} = j)$	13 ∃I
15.	$S\overline{\mathsf{n}} < j$	14 abv
16.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	15 ∨I
17.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	10,11-16 ∃E
18.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	4,5-9,10-17 ∨E
19.	$j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j$	2,3-18 ∃E
20.	$\overline{\mathsf{n}} < j \rightarrow (j = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < j)$	1-19 →I
21.	$\forall x[\overline{\mathsf{n}} < x \rightarrow (x = S\overline{\mathsf{n}} \vee S\overline{\mathsf{n}} < x)]$	20 ∀I

From Q7, either k is zero or it is not. If k is zero, it is a simple addition problem to show that $j = S\overline{\mathsf{n}}$ and so obtain the desired result. If k is a successor, then $S\overline{\mathsf{n}} < j$ and again we have the desired result.

With these theorems in hand, we are ready to obtain the result at which we have been aiming.

T8.19. For any n, (i) Q $\vdash_{ND}$ $\forall x(x \leq \overline{\mathsf{n}} \vee \overline{\mathsf{n}} \leq x)$ and (ii) Q $\vdash_{ND}$ $\forall x(x < \overline{\mathsf{n}} \vee x = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < x)$.

We show (ii). By induction on n we show Q $\vdash_{ND}$ $j < \overline{\mathsf{n}} \vee j = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < j$; the result immediately follows by ∀I.

Basis: $\mathsf{n} = 0$. We need to show that Q $\vdash_{ND}$ $j < \overline{\mathsf{0}} \vee j = \overline{\mathsf{0}} \vee \overline{\mathsf{0}} < j$.

1.	$j = \overline{0} \vee \exists y(j = Sy)$	from Q7
2.	$j = \overline{0}$	A (g 1$\vee$E)
3.	$j = \overline{0} \vee \overline{0} < j$	2 $\vee$I
4.	$\exists y(j = Sy)$	A (g 1$\vee$E)
5.	$j = Sk$	A (g 4$\exists$E)
6.	$Sk + \overline{0} = Sk$	from Q3
7.	$Sk + \overline{0} = j$	6,5 =E
8.	$\exists v(Sv + \overline{0} = j)$	7 $\exists$I
9.	$\overline{0} < j$	8 abv
10.	$j = \overline{0} \vee \overline{0} < j$	9 $\vee$I
11.	$j = \overline{0} \vee \overline{0} < j$	4,5-10 $\exists$E
12.	$j = \overline{0} \vee \overline{0} < j$	1,2-3,4-11 $\vee$E
13.	$j < \overline{0} \vee j = \overline{0} \vee \overline{0} < j$	12 $\vee$I

Assp: For any i, $0 \leq \mathsf{i} < \mathsf{k}$, $\mathsf{Q} \vdash_{ND} j < \overline{\mathsf{i}} \vee j = \overline{\mathsf{i}} \vee \overline{\mathsf{i}} < j$

Show: $\mathsf{Q} \vdash_{ND} j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$

1.	$j < \overline{\mathsf{k}-1} \vee j = \overline{\mathsf{k}-1} \vee \overline{\mathsf{k}-1} < j$	by assumption
2.	$j < \overline{\mathsf{k}-1}$	A (g 1$\vee$E)
3.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{k}-1-1}$	from 2 with T8.16
4.	$\emptyset \neq \emptyset \vee j = \overline{0} \vee \ldots \vee j = \overline{\mathsf{k}-1-1} \vee j = \overline{\mathsf{k}-1}$	3 $\vee$I
5.	$j < \overline{\mathsf{k}}$	from 4 with T8.17
6.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	5 $\vee$I
7.	$j = \overline{\mathsf{k}-1}$	A (g 1$\vee$E)
8.	$\overline{\mathsf{k}-1} < \overline{\mathsf{k}}$	T8.14 ($\mathsf{k}-1 < \mathsf{k}$)
9.	$j < \overline{\mathsf{k}}$	8,7 =E
10.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	9 $\vee$I
11.	$\overline{\mathsf{k}-1} < j$	A (g 1$\vee$E)
12.	$j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	from 11 with T8.18
13.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	12 $\vee$I
14.	$j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$	1, etc. $\vee$E

So $\mathsf{Q} \vdash_{ND} j < \overline{\mathsf{k}} \vee j = \overline{\mathsf{k}} \vee \overline{\mathsf{k}} < j$.

Indct: For any n, $\mathsf{Q} \vdash_{ND} j < \overline{\mathsf{n}} \vee j = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < j$; and the desired result follows by $\forall$I.

Note the use of theorems T8.16, T8.17 and T8.18. In the first case of the show we convert from one inequality to another by switching to an extended disjunction,

adding a disjunct and then converting back to the second inequality. Also again you should be clear about how the extended disjunctions work. If $\mathsf{k}-1 = 0$, then the disjunction at (3) reduces to $\emptyset \neq \emptyset$ and the one at (4) to $\emptyset \neq \emptyset \vee j = \overline{0}$. But this is just why we have been sure that there *is* some formula in these cases, so that the argument continues to work.

E8.33. Complete the demonstration of T8.19 by showing part (i) of T8.17, T8.18 and then T8.19.

8.4.6 Case

Finally, three theorems to round out results about inequality.

T8.20. For any n and formula $\mathcal{P}(x)$, (i) if Q $\vdash_{ND}$ $\mathcal{P}(\overline{0})$ or Q $\vdash_{ND}$ $\mathcal{P}(\overline{1})$ or ... or Q $\vdash_{ND}$ $\mathcal{P}(\overline{\mathsf{n}})$ then Q $\vdash_{ND}$ $(\exists x \leq \overline{\mathsf{n}})\mathcal{P}(x)$, and (ii) if $0 \neq 0$ or Q $\vdash_{ND}$ $\mathcal{P}(\overline{0})$ or ... or Q $\vdash_{ND}$ $\mathcal{P}(\overline{\mathsf{n}-1})$ then Q $\vdash_{ND}$ $(\exists x < \overline{\mathsf{n}})\mathcal{P}(x)$.

In the second case, again, we include the first disjunct to keep the conditional defined in the case when $\mathsf{n} = 0$; then the conditional obtains because the antecedent does not. This theorem is nearly trivial. (i) For some $\mathsf{m} \leq \mathsf{n}$ suppose $\mathcal{P}(\overline{\mathsf{m}})$; by T8.14, Q $\vdash_{ND}$ $\overline{\mathsf{m}} \leq \overline{\mathsf{n}}$; so by ($\exists$I), Q $\vdash_{ND}$ $(\exists x \leq \overline{\mathsf{n}})\mathcal{P}(x)$. Similarly for (ii).

If $\mathcal{P}$ is true of some individual $\leq \mathsf{n}$ or $< \mathsf{n}$ then it is immediate that the corresponding bounded existential generalization is true.

*T8.21. For any n and formula $\mathcal{P}(x)$, (i) if Q $\vdash_{ND}$ $\mathcal{P}(\overline{0})$ and Q $\vdash_{ND}$ $\mathcal{P}(\overline{1})$ and ... and Q $\vdash_{ND}$ $\mathcal{P}(\overline{\mathsf{n}})$ then Q $\vdash_{ND}$ $(\forall x \leq \overline{\mathsf{n}})\mathcal{P}(x)$, and (ii) if $0 = 0$ and Q $\vdash_{ND}$ $\mathcal{P}(\overline{0})$ and ... and Q $\vdash_{ND}$ $\mathcal{P}(\overline{\mathsf{n}-1})$ then Q $\vdash_{ND}$ $(\forall x < \overline{\mathsf{n}})\mathcal{P}(x)$.

This time, in the second case we include a trivial truth in order to keep the conditional defined when $\mathsf{n} = 0$; when $\mathsf{n} = 0$, then the antecedent is trivially true, but the consequent follows from nothing. The argument is by induction on the value of n.

If Q proves $\mathcal{P}$ for each individual $\leq \overline{\mathsf{n}}$ or $< \overline{\mathsf{n}}$ then Q proves the corresponding bounded universal generalization.

*T8.22. For any n, (i) Q $\vdash_{ND}$ $\forall x[x \leq \overline{\mathsf{n}} \leftrightarrow (x < \overline{\mathsf{n}} \vee x = \overline{\mathsf{n}})]$, and (ii) Q $\vdash_{ND}$ $\forall x[x < \overline{\mathsf{n}} \leftrightarrow (x \leq \overline{\mathsf{n}} \wedge x \neq \overline{\mathsf{n}})]$

Hint: You will be able to move between the long disjunctions on the one hand, and inequalities of the different types on the other. Part (i) does not require induction. For (ii), it will be helpful to begin by showing, by induction on a, that for any $\mathsf{a} \leq \mathsf{n}$, $\mathrm{Q} \vdash_{ND} j < \bar{\mathsf{a}} \to j \neq \bar{\mathsf{n}}$ — the case when $\mathsf{a} = \mathsf{n}$ gives $\mathrm{Q} \vdash_{ND} j < \bar{\mathsf{n}} \to j \neq \bar{\mathsf{n}}$.

In the obvious way, we are able to express $s \leq t$ in terms of $s < t$ and similarly, $s < t$ in terms of $s \leq t$.

*E8.34. Provide derivations to show both parts of T8.21.

*E8.35. Provide derivations to show both parts of T8.22.

E8.36. After a few days studying mathematical logic, Zeno hits upon what he thinks is conclusive proof that all is one. He argues, by mathematical induction that all the members of any n-tuple are identical. From this, he considers the n-tuple consisting of you and Mount Rushmore, and concludes that you are identical; similarly for you and G.W. Bush, and so forth. What is the matter with Zeno's reasoning? Hint: Is the reasoning at the show stage truly arbitrary? does it apply to any k?

Basis: If A is a 1-tuple, then it is of the sort $\langle \mathsf{o} \rangle$, and every member of $\langle \mathsf{o} \rangle$ is identical. So every member of A is identical.

Assp: For any i, $1 \leq i < k$, all the members of any i-tuple are identical.

Show: All the members of any k-tuple are identical.

If A is a k-tuple, then it is of the form $\langle \mathsf{o}_1 \ldots \mathsf{o}_{k-2}, \mathsf{o}_{k-1}, \mathsf{o}_k \rangle$. But both $\langle \mathsf{o}_1 \ldots \mathsf{o}_{k-2}, \mathsf{o}_{k-1} \rangle$ and $\langle \mathsf{o}_1 \ldots \mathsf{o}_{k-2}, \mathsf{o}_k \rangle$ are $k-1$ tuples; so by the inductive assumption, all their members are identical; but these have o_1 in common and together include all the members of A; so all the members of A are identical to o_1 and so to one another.

Indct: All the members of any A are identical.

E8.37. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where

it does not. Your essay should exhibit an understanding of methods from the text.

a. The use of the inductive assumption in an argument from mathematical induction.

b. The reason mathematical induction works as a deductive argument form.

Final Theorems of Chapter 8

T8.8 For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a} + \mathsf{b} = \mathsf{c}$, then $\mathsf{Q} \vdash_{ND} \bar{\mathsf{a}} + \bar{\mathsf{b}} = \bar{\mathsf{c}}$.

T8.9 For any $\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{U}$, if $\mathsf{a} \times \mathsf{b} = \mathsf{c}$ then $\mathsf{Q} \vdash_{ND} \bar{\mathsf{a}} \times \bar{\mathsf{b}} = \bar{\mathsf{c}}$.

T8.10 For any $\mathsf{a}, \mathsf{b} \in \mathsf{U}$, if $\mathsf{a} \neq \mathsf{b}$, then $\mathsf{Q} \vdash_{ND} \bar{\mathsf{a}} \neq \bar{\mathsf{b}}$

T8.11 $\mathsf{Q} \vdash_{ND} Sj + \bar{\mathsf{n}} = j + S\bar{\mathsf{n}}$.

T8.12 (i) If $\mathsf{a} \not\leq \mathsf{b}$, then $\mathsf{Q} \vdash_{ND} \bar{\mathsf{a}} \not\leq \bar{\mathsf{b}}$; and (ii) If $\mathsf{a} \not< \mathsf{b}$, then $\mathsf{Q} \vdash_{ND} \bar{\mathsf{a}} \not< \bar{\mathsf{b}}$.

T8.13 For any variable-free term t of $\mathcal{L}_{NT}$, if $\mathsf{N}[t] = \mathsf{n}$, then $\mathsf{Q} \vdash_{ND} t = \bar{\mathsf{n}}$

T8.14 Q correctly decides atomic sentences of $\mathcal{L}_{NT}$. For any sentence $\mathcal{P}$ of the sort $s = t$, $s \leq t$ or $s < t$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then $\mathsf{Q} \vdash_{ND} \mathcal{P}$; and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then $\mathsf{Q} \vdash_{ND} \sim\mathcal{P}$.

T8.15 For any n and T, if $T \vdash_{ND} x = Sy$ and $T \vdash_{ND} y = \bar{0} \vee y = \bar{1} \vee \ldots \vee y = \bar{\mathsf{n}}$, then $T \vdash_{ND} x = S\bar{0} \vee x = S\bar{1} \vee \ldots \vee x = S\bar{\mathsf{n}}$

T8.16 For any n, (i) $\mathsf{Q} \vdash_{ND} (\forall x \leq \bar{\mathsf{n}})(x = \bar{0} \vee x = \bar{1} \ldots \vee x = \bar{\mathsf{n}})$ and (ii) $\mathsf{Q} \vdash_{ND} (\forall x < \bar{\mathsf{n}})(\emptyset \neq \emptyset \vee x = \bar{0} \vee x = \bar{1} \ldots \vee x = \overline{\mathsf{n}-1})$

T8.17 For any n, (i) $\mathsf{Q} \vdash_{ND} \forall x([x = \bar{0} \vee x = \bar{1} \ldots \vee x = \bar{\mathsf{n}}] \rightarrow x \leq \bar{\mathsf{n}})$ and (ii) $\mathsf{Q} \vdash_{ND} \forall x([\emptyset \neq \emptyset \vee x = \bar{0} \vee \ldots \vee x = \overline{\mathsf{n}-1}] \rightarrow x < \bar{\mathsf{n}})$

T8.18 For any n, (i) $\mathsf{Q} \vdash_{ND} \forall x[\bar{\mathsf{n}} \leq x \rightarrow (\bar{\mathsf{n}} = x \vee S\bar{\mathsf{n}} \leq x)]$ and (ii) $\forall x[\bar{\mathsf{n}} < x \rightarrow (S\bar{\mathsf{n}} = x \vee S\bar{\mathsf{n}} < x)]$

T8.19 For any n, (i) $\mathsf{Q} \vdash_{ND} \forall x(x \leq \bar{\mathsf{n}} \vee \bar{\mathsf{n}} \leq x)$ and (ii) $\mathsf{Q} \vdash_{ND} \forall x(x < \bar{\mathsf{n}} \vee x = \bar{\mathsf{n}} \vee \bar{\mathsf{n}} < x)$

T8.20 For any n and formula $\mathcal{P}(x)$, (i) if $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{0})$ or $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{1})$ or ... or $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{\mathsf{n}})$ then $\mathsf{Q} \vdash_{ND} (\exists x \leq \bar{\mathsf{n}})\mathcal{P}(x)$, and (ii) if $0 \neq 0$ or $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{0})$ or ... or $\mathsf{Q} \vdash_{ND} \mathcal{P}(\overline{\mathsf{n}-1})$ then $\mathsf{Q} \vdash_{ND} (\exists x < \bar{\mathsf{n}})\mathcal{P}(x)$.

T8.21 For any n and formula $\mathcal{P}(x)$, (i) if $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{0})$ and $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{1})$ and ... and $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{\mathsf{n}})$ then $\mathsf{Q} \vdash_{ND} (\forall x \leq \bar{\mathsf{n}})\mathcal{P}(x)$, and (ii) if $0 = 0$ and $\mathsf{Q} \vdash_{ND} \mathcal{P}(\bar{0})$ and ... and $\mathsf{Q} \vdash_{ND} \mathcal{P}(\overline{\mathsf{n}-1})$ then $\mathsf{Q} \vdash_{ND} (\forall x < \bar{\mathsf{n}})\mathcal{P}(x)$.

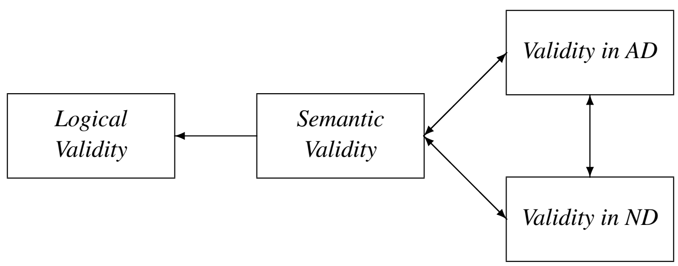
T8.22 For any n, (i) $\mathsf{Q} \vdash_{ND} \forall x[x \leq \bar{\mathsf{n}} \leftrightarrow (x < \bar{\mathsf{n}} \vee x = \bar{\mathsf{n}})]$, and (ii) $\mathsf{Q} \vdash_{ND} \forall x[x < \bar{\mathsf{n}} \leftrightarrow (x \leq \bar{\mathsf{n}} \wedge x \neq \bar{\mathsf{n}})]$

Part III

Classical Metalogic: Soundness and Adequacy

Introductory

In Part I we introduced four notions of validity. In this part, we set out to show that they are interrelated as follows.



An argument is semantically valid iff it is valid in the derivation systems. So the three formal notions apply to exactly the same arguments. And if an argument is semantically valid, then it is logically valid. So any of the formal notions imply logical validity for a corresponding ordinary argument.

More carefully, in Part I, we introduced four main notions of validity. There are logical validity from chapter 1, semantic validity from chapter 4, and syntactic validity in the derivation systems *AD*, from chapter 3 and *ND* from chapter 6. We turn in this part to the task of thinking *about* these notions, and especially about how they are related. The primary result is that $\Gamma \vDash \mathcal{P}$ iff $\Gamma \vdash_{AD} \mathcal{P}$ iff $\Gamma \vdash_{ND} \mathcal{P}$ (iff $\Gamma \vdash_{ND+} \mathcal{P}$). Thus our different formal notions of validity are met by just the same arguments, and the derivation systems — themselves defined in terms of *form* are "faithful" to the semantic notion: what is derivable is neither more nor less than what is semantically valid. And this is just right: If what is derivable were more than what is semantically valid, derivations could lead us from true premises to false conclusions; if it were less, not all semantically valid arguments could be identified as such by derivations. That the derivable is no *more* than what is semantically valid, is known as *soundness* of a derivation system; that it is no *less* is *adequacy*. In addition,

we show that if an argument is semantically valid, then a corresponding ordinary argument is *logically valid*. Given the equivalence between the formal notions of validity, it follows that if an argument is valid in any of the formal senses, then it is logically valid. This connects the formal machinery to the notion of validity with which we began.[2]

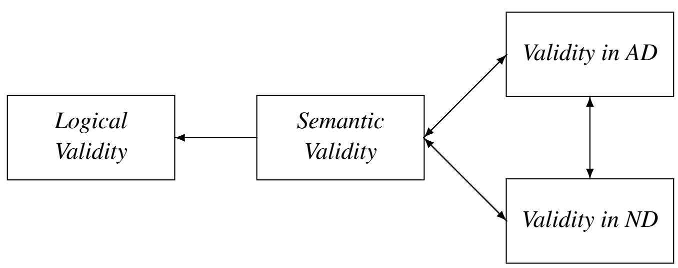
We begin in chapter 9 showing that just the same arguments are valid in the derivation systems *ND* and *AD*. This puts us in a position to demonstrate in chapter 10 the core result that the derivation systems are both sound and adequate. Chapter chapter 11 fills out this core picture in different directions.

[2] *Adequacy* is commonly described as *completeness*. However, this only invites confusion with theory completeness as described in Part IV.

Chapter 9

Preliminary Results

We have said that the aim of this part is to establish the following relations: An argument is semantically valid iff it is valid in *AD*; iff it is valid in *ND*; and if an argument is semantically valid, then it is logically valid.



In this chapter, we begin to develop these relations, taking up some of the simpler cases. We consider the leftmost horizontal arrow, and the rightmost vertical ones. Thus we show that quantificational (semantic) validity implies logical validity, that validity in *AD* implies validity in *ND*, and that validity in *ND* implies validity in *AD* (and similarly for *ND+*). Implications between semantic validity and the syntactical notions will wait for chapter 10.

9.1 Semantic Validity Implies Logical Validity

Logical validity is defined for arguments in ordinary language. From LV, an argument is logically valid iff there is no consistent *story* in which all the premises are true and the conclusion is false. Quantificational validity is defined for arguments in

a formal language. From QV, an argument is quantificationally valid iff there is no *interpretation* on which all the premises are true and the conclusion is not. So our task is to show how facts about formal expressions and interpretations connect with ordinary expressions and stories. In particular, where $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is an ordinary-language argument, and $\mathcal{P}'_1 \ldots \mathcal{P}'_n$, $\mathcal{Q}'$ are the formulas of a good translation, we show that if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'_1$, then the ordinary argument $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is logically valid. The reasoning itself is straightforward. We will spend a bit more time discussing the result.

Recall our criterion of goodness for translation CG from chapter 5 (p. 138). When we identify an interpretation function II (sentential or quantificational), we thereby identify an *intended interpretation* II_ω corresponding to any way ω that the world can be. For example, corresponding to the interpretation function,

II B: Bill is happy
H: Hill is happy

$\mathsf{II}_\omega[B] = \mathsf{T}$ just in case Bill is happy at ω, and similarly for H. Given this, a formal translation $\mathcal{A}'$ of some ordinary $\mathcal{A}$ is *good* only if at any ω, $\mathsf{II}_\omega[\mathcal{A}']$ has the same truth value as $\mathcal{A}$ at ω. Given this, we can show,

T9.1. For any ordinary argument $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$, with good translation consisting of II and $\mathcal{P}'_1 \ldots \mathcal{P}'_n$, $\mathcal{Q}'$, if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$, then $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is logically valid.

Suppose $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$ but $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is not logically valid. From the latter, by LV, there is some consistent story where each of $\mathcal{P}_1 \ldots \mathcal{P}_n$ is true but $\mathcal{Q}$ is false. Since $\mathcal{P}_1 \ldots \mathcal{P}_n$ are true at ω, by CG, $\mathsf{II}_\omega[\mathcal{P}'_1] = \mathsf{T}$, and ... and $\mathsf{II}_\omega[\mathcal{P}'_n] = \mathsf{T}$. And since ω is consistent with $\mathcal{Q}$ false at ω, $\mathcal{Q}$ is not both true and false at ω; so $\mathcal{Q}$ is not true at ω; so by by CG, $\mathsf{II}_\omega[\mathcal{Q}'_1] \neq \mathsf{T}$. So there is an I that that makes each of $\mathsf{I}[\mathcal{P}'_1] = \mathsf{T}$, and ... and $\mathsf{I}[\mathcal{P}'_n] = \mathsf{T}$ and $\mathsf{I}[\mathcal{Q}'] \neq \mathsf{T}$; so by QV, $\mathcal{P}'_1 \ldots \mathcal{P}'_n \nvDash \mathcal{Q}'$. This is impossible; reject the assumption: if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$ then $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is logically valid.

It is that easy. If there is no interpretation where $\mathcal{P}'_1 \ldots \mathcal{P}'_n$ are true but $\mathcal{Q}'$ is not, then there is no *intended* interpretation where $\mathcal{P}'_1 \ldots \mathcal{P}'_n$ are true but $\mathcal{Q}'$ is not; so, by CG, there is no consistent story where the premises are true and the conclusion is not; so $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$, is logically valid. So if $\mathcal{P}'_1 \ldots \mathcal{P}'_n \vDash \mathcal{Q}'$ then $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is logically valid.

Let us make a couple of observations: First, CG is stronger than is actually required for our application of semantic to logical validity. CG requires a biconditional for good translation.

$$\omega \quad \rightleftarrows \quad \mathsf{I\!I}_\omega$$

$\mathcal{A}$ is true at ω iff $\mathsf{I\!I}_\omega[\mathcal{A}'] = \mathsf{T}$. But our reasoning applies to premises just the left-to-right portion of this condition: if $\mathcal{P}$ is true at ω then $\mathsf{I\!I}_\omega[\mathcal{P}'] = \mathsf{T}$. And for the conclusion, the reasoning goes in the opposite direction: if $\mathsf{I\!I}_\omega[\mathcal{Q}'] = \mathsf{T}$ then $\mathcal{Q}$ is true at ω (so that if the consequent fails at ω, then the antecedent fails at $\mathsf{I\!I}_\omega$). The biconditional from CG guarantees both. But, strictly, for premises, all we need is that truth of an ordinary expression at a story guarantees truth for the corresponding formal one at the intended interpretation. And for a conclusion, all we need is that truth of the formal expression on the intended interpretation guarantees truth of the corresponding ordinary expression at the story.

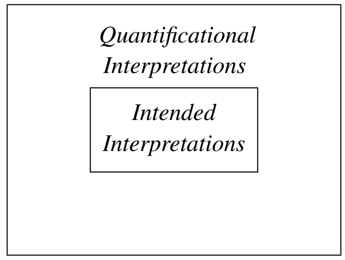
Thus we might use our methods to identify logical validity even where translations are less than completely good. Consider, for example, the following argument.

(A) Bob took a shower and got dressed
    ———
    Bob took a shower

As discussed in chapter 5 (p. 156), where $\mathsf{I\!I}$ gives S the same value as "Bob took a shower" and D the same as "Bob got dressed," we might agree that there are cases where $\mathsf{I\!I}_\omega[S \wedge D] = \mathsf{T}$ but "Bob took a shower and got dressed" is false. So we might agree that the right-to-left conditional is false, and the translation is not good.

However, even if this is so, given our interpretation function, there is no situation where "Bob took a shower and got dressed" is true but $S \wedge D$ is F at the corresponding intended interpretation. So the left-to-right conditional is sustained. So, even if the translation is not good by CG, it remains possible to use our methods to demonstrate logical validity. Since it remains that if the ordinary premise is true at a story, then the formal expression is true at the corresponding intended interpretation, semantic validity implies logical validity. A similar point applies to conclusions. Of course, we already knew that this argument is logically valid. But the point applies to more complex arguments as well.

Second, observe that our reasoning does not work in reverse. It might be that $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is logically valid, even though $\mathcal{P}'_1 \ldots \mathcal{P}'_n \not\vDash \mathcal{Q}'$. Finding a quantificational interpretation where $\mathcal{P}'_1 \ldots \mathcal{P}'_n$ are true and $\mathcal{Q}'$ is not shows that $\mathcal{P}'_1 \ldots \mathcal{P}'_n \not\vDash \mathcal{Q}'$. However it does not show that $\mathcal{P}_1 \ldots \mathcal{P}_n / \mathcal{Q}$ is not logically valid. Here is why: There may be quantificational interpretations which do not correspond to any consistent story. The situation is like this:



Intended interpretations correspond to stories. If no interpretation whatsoever has the premises true and the conclusion not, then no intended interpretation has the premises true and conclusion not, so no consistent story makes the premises true and the conclusion not. But it may be that some (unintended) interpretation makes the premises true and conclusion false, even though no intended interpretation is that way. Thus, if we were to attempt to run the above reasoning in reverse, a move from the assumption that $\mathcal{P}'_1 \dots \mathcal{P}'_n \not\models \mathcal{Q}'$, to the conclusion that there is a consistent story where $\mathcal{P}_1 \dots \mathcal{P}_n$ are true but $\mathcal{Q}$ is not, would fail.

It is easy to see why there might be unintended interpretations. Consider, first, this standard argument.

(B)
All humans are mortal
Socrates is human
———
Socrates is mortal

It is logically valid. But consider what happens when we translate into a *sentential* language. We might try an interpretation function as follows.

A: All humans are mortal

H: Socrates is human

M: Socrates is mortal

with translation, A, H / M. But, of course, there is a row of the truth table on which A and H are T and M is F. So the argument is not sententially valid. This interpretation is unintended in the sense that it corresponds to no consistent story whatsoever. Sentential languages are sufficient to identify validity when validity results from truth functional structure; but this argument is not valid because of truth functional structure.

We are in a position to expose its validity only in the quantificational case. Thus we might have,

s: Socrates

H^1: {o | o is human}

M^1: {o | o is mortal}

with translation $\forall x(Hx \rightarrow Mx)$, Hs / Ms. The argument is quantificationally valid. And, as above, it follows that the ordinary one is logically valid.

But related problems may arise even for quantificational languages. Thus, consider,

(C) $\dfrac{\text{Socrates is necessarily human}}{\text{Socrates is human}}$

Again, the argument is logically valid. But now we end up with something like an additional relation symbol N^1 for {o | o is necessarily human}, and translation Ns/Hs. And this is not quantificationally valid. Consider, for example, an interpretation with $\mathsf{U} = \{1\}$, $\mathsf{I}[s] = 1$, $\mathsf{I}[N] = \{1\}$, and $\mathsf{I}[H] = \{\}$. Then the premise is true, but the conclusion is not. Again, the interpretation corresponds to no consistent story. And, again, the argument includes structure that our quantificational language fails to capture. As it turns out, *modal* logic is precisely an attempt to work with structure introduced by notions of possibility and necessity. Where '$\Box$' represents necessity, this argument, with translation $\Box Hs/Hs$ is valid on standard modal systems.

The upshot of this discussion is that our methods are adequate when they work to identify validity. When an argument is semantically valid, we can be sure that it is logically valid. But we are not in a position to identify all the arguments that are logically valid. Thus quantificational invalidity does not imply logical invalidity. We should not be discouraged by this or somehow put off the logical project. Rather, we have a rationale for *expanding* the logical project! In Part I, we set up formal logic as a "tool" or "machine" to identify logical validity. Beginning with the notion of logical validity, we introduce our formal languages, learn to translate into them, and to manipulate arguments by semantical and syntactical methods. The sentential notions have some utility. But when it turns out that sentential languages miss important structure, we expand the language to include quantificational structure, developing the semantical and syntactical methods to match. And similarly, if our quantificational languages should turn out to miss important structure, we expand the language to capture that structure, and further develop the semantical and syntactical methods. As it happens, the classical quantificational logic we have so far seen is sufficient to identify validity in a wide variety of contexts — and, in particular, for arguments in

mathematics. Also, controversy may be introduced as one expands beyond the classical quantificational level. So the logical project is a live one. But let us return to the kinds of validity we have already seen.

E9.1. (i) Recast the above reasoning to show directly a corollary to T9.1: If $\vDash \mathcal{Q}'$, then $\mathcal{Q}$ is necessarily true (that is, true in any consistent story). (ii) Suppose $\nvDash \mathcal{Q}'$; does it follow that $\mathcal{Q}$ is not necessary (that is, not true in some consistent story)? Explain.

9.2 Validity in *AD* Implies Validity in *ND*

It is easy to see that if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$. Roughly, anything we can accomplish in *AD*, we can accomplish in *ND* as well. If a premise appears in an *AD* derivation, that same premise can be used in *ND*. If an axiom appears in an *AD* derivation, that axiom can be derived in *ND*. And if a line is justified by MP or Gen in *AD*, that same line may be justified by rules of *ND*. So anything that can be derived in *AD* can be derived in *ND*. Officially, this reasoning is by induction on the line numbers of an *AD* derivation, and it is appropriate to work out the details more formally. The argument by mathematical induction is longer than anything we have seen so far, but the reasoning is straightforward.

T9.2. If $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

Suppose $\Gamma \vdash_{AD} \mathcal{P}$. Then there is an *AD* derivation $A = \langle \mathcal{Q}_1 \ldots \mathcal{Q}_n \rangle$ of $\mathcal{P}$ from premises in Γ, with $\mathcal{Q}_n = \mathcal{P}$. We show that there is a corresponding *ND* derivation N, such that if $\mathcal{Q}_i$ appears on line i of A, then $\mathcal{Q}_i$ appears, under the scope of the premises alone, on the line numbered 'i' of N. It follows that $\Gamma \vdash_{ND} \mathcal{P}$. For any *premises* $\mathcal{Q}_a$, $\mathcal{Q}_b$,... $\mathcal{Q}_j$ in A, let N begin,

0.a	$\mathcal{Q}_a$	P
0.b	$\mathcal{Q}_b$	P
$\vdots$		
0.j	$\mathcal{Q}_j$	P

Now we reason by induction on the line numbers in A. The general plan is to *construct* a derivation N which accomplishes just what is accomplished in A. Fractional line numbers, as above, maintain the parallel between the two derivations.

Basis: $\mathcal{Q}_1$ in A is a premise or an instance of A1, A2, A3, A4, A5, A6 or A7.

(prem) If $\mathcal{Q}_1$ is a premise $\mathcal{Q}_i$, continue N as follows,

$$\begin{array}{r|ll}
0.a & \mathcal{Q}_a & \text{P} \\
0.b & \mathcal{Q}_b & \text{P} \\
\vdots & & \\
0.j & \mathcal{Q}_j & \text{P} \\
1 & \mathcal{Q}_i & \text{0.i R}
\end{array}$$

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered '1' of N.

(A1) If $\mathcal{Q}_1$ is an instance of A1, then it is of the form, $\mathcal{B} \to (\mathcal{C} \to \mathcal{B})$, and we continue N as follows,

$$\begin{array}{r|ll}
0.a & \mathcal{Q}_a & \text{P} \\
0.b & \mathcal{Q}_b & \text{P} \\
\vdots & & \\
0.j & \mathcal{Q}_j & \text{P} \\
1.1 & \quad\vert\ \underline{\mathcal{B}} & \text{A } (g, \to\text{I}) \\
1.2 & \quad\vert\quad\vert\ \underline{\mathcal{C}} & \text{A } (g, \to\text{I}) \\
1.3 & \quad\vert\quad\vert\ \mathcal{B} & \text{1.1 R} \\
1.4 & \quad\vert\ \mathcal{C} \to \mathcal{B} & \text{1.2-1.3} \to\text{I} \\
1 & \mathcal{B} \to (\mathcal{C} \to \mathcal{B}) & \text{1.1-1.4} \to\text{I}
\end{array}$$

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered '1' of N.

(A2) If $\mathcal{Q}_1$ is an instance of A2, then it is of the form, $(\mathcal{B} \to (\mathcal{C} \to \mathcal{D})) \to ((\mathcal{B} \to \mathcal{C}) \to (\mathcal{B} \to \mathcal{D}))$ and we continue N as follows,

$$\begin{array}{r|ll}
0.a & \mathcal{Q}_a & \text{P} \\
0.b & \mathcal{Q}_b & \text{P} \\
\vdots & & \\
0.j & \mathcal{Q}_j & \text{P} \\
1.1 & \quad\vert\ \underline{\mathcal{B} \to (\mathcal{C} \to \mathcal{D})} & \text{A } (g, \to\text{I}) \\
1.2 & \quad\vert\quad\vert\ \underline{\mathcal{B} \to \mathcal{C}} & \text{A } (g, \to\text{I}) \\
1.3 & \quad\vert\quad\vert\quad\vert\ \underline{\mathcal{B}} & \text{A } (g, \to\text{I}) \\
1.4 & \quad\vert\quad\vert\quad\vert\ \mathcal{C} & \text{1.2,1.3} \to\text{E} \\
1.5 & \quad\vert\quad\vert\quad\vert\ \mathcal{C} \to \mathcal{D} & \text{1.1,1.3} \to\text{E} \\
1.6 & \quad\vert\quad\vert\quad\vert\ \mathcal{D} & \text{1.5,1.4} \to\text{E} \\
1.7 & \quad\vert\quad\vert\ \mathcal{B} \to \mathcal{D} & \text{1.3-1.6} \to\text{I} \\
1.8 & \quad\vert\ (\mathcal{B} \to \mathcal{C}) \to (\mathcal{B} \to \mathcal{D}) & \text{1.2-1.7} \to\text{I} \\
1 & (\mathcal{B} \to (\mathcal{C} \to \mathcal{D})) \to ((\mathcal{B} \to \mathcal{C}) \to (\mathcal{B} \to \mathcal{D})) & \text{1.1-1.8} \to\text{I}
\end{array}$$

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered '1' of N.

(A3) Homework.

(A4) If $\mathcal{Q}_1$ is an instance of A4, then it is of the form $\forall x \mathcal{B} \to B^x_t$ for some variable x and term t that is free for x in $\mathcal{B}$, and we continue N as follows,

0.a	$\mathcal{Q}_a$	P
0.b	$\mathcal{Q}_b$	P
$\vdots$		
0.j	$\mathcal{Q}_j$	P
1.1	$\quad \forall x \mathcal{B}$	A ($g, \to$I)
1.2	$\quad \mathcal{B}^x_t$	1.1 $\forall$E
1	$\forall x \mathcal{B} \to \mathcal{B}^x_t$	1.1-1.2 $\to$I

Since we are given that t is free for x in $\mathcal{B}$, the parallel requirement on $\forall$E is met at line 1.2. So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered '1' of N.

(A5) Homework.

(A6) If $\mathcal{Q}_1$ is an instance of A6, then it is of the form $(x_i = y) \to (h^n x_1 \dots x_i \dots x_n = h^n x_1 \dots y \dots x_n)$ for some variables $x_1 \dots x_n$ and y and function symbol h^n; and we continue N as follows,

0.a	$\mathcal{Q}_a$	P
0.b	$\mathcal{Q}_b$	P
$\vdots$		
0.j	$\mathcal{Q}_j$	P
1.1	$\quad x_i = y$	A ($g, \to$I)
1.2	$\quad h^n x_1 \dots x_i \dots x_n = h^n x_1 \dots x_i \dots x_n$	=I
1.3	$\quad h^n x_1 \dots x_i \dots x_n = h^n x_1 \dots y \dots x_n$	1.2,1.1 =E
1	$(x_i = y) \to (h^n x_1 \dots x_i \dots x_n = h^n x_1 \dots y \dots x_n)$	1.1-1.3 $\to$I

So $\mathcal{Q}_1$ appears, under the scope of the premises alone, on the line numbered '1' of N.

(A7) Homework.

Assp: For any i, $1 \leq i < k$, if $\mathcal{Q}_i$ appears on line i of A, then $\mathcal{Q}_i$ appears, under the scope of the premises alone, on the line numbered 'i' of N.

Show: If $\mathcal{Q}_k$ appears on line k of A, then $\mathcal{Q}_k$ appears, under the scope of the premises alone, on the line numbered 'k' of N.

$\mathcal{Q}_k$ in A is a premise, an axiom, or arises from previous lines by MP or Gen. If $\mathcal{Q}_k$ is a premise or an axiom then, by reasoning as in the basis (with line numbers adjusted to $k.n$) if $\mathcal{Q}_k$ appears on line k of A, then $\mathcal{Q}_k$ appears, under the scope of the premises alone, on the line numbered 'k' of A. So suppose $\mathcal{Q}_k$ arises by MP or Gen.

(MP) If $\mathcal{Q}_k$ arises from previous lines by MP, then A is as follows,

$$\begin{array}{lll} i & \mathcal{B} & \\ & \vdots & \\ j & \mathcal{B} \to \mathcal{C} & \\ & \vdots & \\ k & \mathcal{C} & i, j \text{ MP} \end{array}$$

where $i, j < k$ and $\mathcal{Q}_k$ is $\mathcal{C}$. By assumption, then, there are lines in N,

$$\begin{array}{l|l} i & \mathcal{B} \\ & \vdots \\ j & \mathcal{B} \to \mathcal{C} \end{array}$$

So we simply continue derivation N,

$$\begin{array}{l|ll} i & \mathcal{B} & \\ & \vdots & \\ j & \mathcal{B} \to \mathcal{C} & \\ & \vdots & \\ k & \mathcal{C} & i, j \to\text{E} \end{array}$$

So $\mathcal{Q}_k$ appears under the scope of the premises alone, on the line numbered 'k' of N.

(Gen) If $\mathcal{Q}_k$ arises from previous lines by Gen, then A is as follows,

$$\begin{array}{lll} i & \mathcal{B} \to \mathcal{C} & \\ & \vdots & \\ k & \mathcal{B} \to \forall x \mathcal{C} & i \text{ Gen} \end{array}$$

where $i < k$, variable x is not free in $\mathcal{B}$, and $\mathcal{Q}_k$ is $\mathcal{B} \to \forall x \mathcal{C}$. By assumption N has a line i,

$$\begin{array}{l|l} & \vdots \\ i & \mathcal{B} \to \mathcal{C} \\ & \vdots \end{array}$$

under the scope of the premises alone. So we continue N as follows,

$$
\begin{array}{r|lll}
i & \mathscr{B} \to \mathscr{C} & & \\
 & \vdots & & \\
k.1 & \quad \mathscr{B} & & \text{A } (g, \to\text{I}) \\
k.2 & \quad \mathscr{C} & & i, k.1 \to\text{E} \\
k.3 & \quad \forall x \mathscr{C} & & k.2\ \forall\text{I} \\
k & \mathscr{B} \to \forall x \mathscr{C} & & k.1\text{-}k.3 \to\text{I}
\end{array}
$$

Since $k.1$ is the only undischarged assumption, and we are given that x is not free in $\mathscr{B}$, x is not free in any undischarged assumption. Further, since there is no change of variables, we can be sure that x is free for every free instance of x in $\mathscr{C}$, and that x is not free in $\forall x \mathscr{C}$. So the restrictions are met on $\forall$I at line $k.3$. So $\mathscr{Q}_k$ appears under the scope of the premises alone, on the line numbered 'k' of N.

In any case then, $\mathscr{Q}_k$ appears under the scope of the premises alone, on the line numbered 'k' of N.

Indct: For any line j of A, $\mathscr{Q}_j$ appears under the scope of the premises alone, on the line numbered 'j' of N.

So $\Gamma \vdash_{ND} \mathscr{Q}_n$, where this is just to say $\Gamma \vdash_{ND} \mathscr{P}$. So T9.2, if $\Gamma \vdash_{AD} \mathscr{P}$, then $\Gamma \vdash_{ND} \mathscr{P}$. Notice the way we use line numbers, $i.1, i.2, \ldots i.n, i$ in N to make good on the claim that for each $\mathscr{Q}_i$ in A, $\mathscr{Q}_i$ appears on the line numbered 'i' of N — where the line numbered 'i' may or may not be the ith line of N. We need this parallel between the line numbers when it comes to cases for MP and Gen. With the parallel, we are in a position to make use of line numbers from justifications in derivation A, directly in the specification of derivation N.

Given an *AD* derivation, what we have done shows that there exists an *ND* derivation, by showing how to construct it. We can see into how this works, by considering an application. Thus, for example, consider the derivation of T3.2 on p. 73.

(D)

1.	$\mathscr{B} \to \mathscr{C}$	prem
2.	$(\mathscr{B} \to \mathscr{C}) \to [\mathscr{A} \to (\mathscr{B} \to \mathscr{C})]$	A1
3.	$\mathscr{A} \to (\mathscr{B} \to \mathscr{C})$	1,2 MP
4.	$[\mathscr{A} \to (\mathscr{B} \to \mathscr{C})] \to [(\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C})]$	A2
5.	$(\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C})$	3,4 MP
6.	$\mathscr{A} \to \mathscr{B}$	prem
7.	$\mathscr{A} \to \mathscr{C}$	5,6 MP

Let this be derivation A; we will follow the method of our induction to construct a corresponding *ND* derivation N. The first step is to list the premises.

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.2	$\mathcal{A} \to \mathcal{B}$	P

Now to the induction itself. The first line of A is a premise. Looking back to the basis case of the induction, we see that we are instructed to produce the line numbered '1' by reiteration. So that is what we do.

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.2	$\mathcal{A} \to \mathcal{B}$	P
1	$\mathcal{B} \to \mathcal{C}$	0.1 R

This may strike you as somewhat pointless! But, again, we need $\mathcal{B} \to \mathcal{C}$ on the line numbered '1' in order to maintain the parallel between the derivations. So our recipe requires this simple step.

Line 2 of A is an instance of A1, and the induction therefore tells us to get it "by reasoning as in the basis." Looking then to the case for A1 in the basis, we continue on that pattern as follows,

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.2	$\mathcal{A} \to \mathcal{B}$	P
1	$\mathcal{B} \to \mathcal{C}$	0.1 R
2.1	$\mid \mathcal{B} \to \mathcal{C}$	A (g, $\to$I)
2.2	$\mid \mid \mathcal{A}$	A (g, $\to$I)
2.3	$\mid \mid \mathcal{B} \to \mathcal{C}$	2.1 R
2.4	$\mid \mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2-2.3 $\to$I
2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	2.1-2.4 $\to$I

Notice that this reasoning for the show step now applies to line 2, so that the line numbers are 2.1, 2.2, 2.3, 2.4, 2 instead of 1.1, 1.2, 1.3, 1.4, 1 as for the basis. Also, what we have added follows *exactly* the pattern from the recipe in the induction, given the relevant instance of A1.

Line 3 is justified by 1,2 MP. Again, by the recipe from the induction, we continue,

0.1	$\mathcal{B} \to \mathcal{C}$	P
0.2	$\mathcal{A} \to \mathcal{B}$	P
1	$\mathcal{B} \to \mathcal{C}$	0.1 R
2.1	$\mid \mathcal{B} \to \mathcal{C}$	A (g, $\to$I)
2.2	$\mid \mid \mathcal{A}$	A (g, $\to$I)
2.3	$\mid \mid \mathcal{B} \to \mathcal{C}$	2.1 R
2.4	$\mid \mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2-2.3 $\to$I
2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	2.1-2.4 $\to$I
3	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	1,2 $\to$E

Notice that the line numbers of the justification are identical to those in the justification from A. And similarly, we are in a position to generate each line in A. Thus, for example, line 4 of A is an instance of A2. So we would continue with lines 4.1-4.8 and 4 to generate the appropriate instance of A2. And so forth. As it turns out, the resultant *ND* derivation is not very efficient! But it is a derivation, and our point is merely to show that some *ND* derivation of the same result exists. So if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

*E9.2. Set up the above induction for T9.2, and complete the unfinished cases to show that if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$. For cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

E9.3. (i) Where A is the derivation for T3.2, complete the process of finding the corresponding derivation N. Hint: if you follow the recipe correctly, the result should have exactly 21 lines. (ii) This derivation N is not very efficient! See if you can find an *ND* derivation to show $\mathcal{A} \rightarrow \mathcal{B}, \mathcal{B} \rightarrow \mathcal{C} \vdash_{ND} \mathcal{A} \rightarrow \mathcal{C}$ that takes fewer than 10 lines.

E9.4. Consider the axiomatic system *A3* as described for E8.11 on p. 394, and produce a complete demonstration that if $\Gamma \vdash_{A3} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

9.3 Validity in *ND* Implies Validity in *AD*

Perhaps the result we have just attained is obvious: if $\Gamma \vdash_{AD} \mathcal{P}$, then of course $\Gamma \vdash_{ND} \mathcal{P}$. But the other direction may be less obvious. Insofar as *AD* may seem to have fewer resources than *ND*, one might wonder whether it is the case that if $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. But, in fact, it is possible to do in *AD* whatever can be done in *ND*. To show this, we need a couple of preliminary results. I begin with an important result known as the *deduction theorem*, turn to some substitution theorems, and finally to the intended result that whatever is provable in *ND* is provable in *AD*.

9.3.1 Deduction Theorem

According to the deduction theorem — subject to an important restriction — if there is an *AD* derivation of $\mathcal{Q}$ from the members of some set of sentences Δ plus $\mathcal{P}$, then there is an *AD* derivation of $\mathcal{P} \rightarrow \mathcal{Q}$ from the members of Δ alone: if $\Delta \cup \{\mathcal{P}\} \vdash_{AD} \mathcal{Q}$ then $\Delta \vdash_{AD} \mathcal{P} \rightarrow \mathcal{Q}$. In practice, this lets us reason just as we do with $\rightarrow$I.

(E)

$$\begin{array}{l|l}
 & \text{members of}\Delta \\
\text{a.} & \quad \mathscr{P} \\
\text{b.} & \quad \mathscr{Q} \\
\text{c.} & \mathscr{P} \to \mathscr{Q} \qquad\qquad \text{a-b deduction theorem}
\end{array}$$

At (b), there is a derivation of $\mathscr{Q}$ from the mbembers of Δ plus $\mathscr{P}$. At (c), the assumption is discharged to indicate a derivation of $\mathscr{P} \to \mathscr{Q}$ from the members of Δ alone. By the deduction theorem, if there is a derivation of $\mathscr{Q}$ from Δ plus $\mathscr{P}$, then there is a derivation of $\mathscr{P} \to \mathscr{Q}$ from Δ alone. Here is the restriction: The discharge of an auxiliary assumption $\mathscr{P}$ is legitimate just in case no application of Gen under its scope generalizes on a variable free in $\mathscr{P}$. The effect is like that of the *ND* restriction on $\forall$I — here, though, the restriction is not on Gen, but rather on the discharge of auxiliary assumptions. In the one case, an assumption available for discharge is one such that no application of Gen under its scope is to a variable free in the assumption; in the other, we cannot apply $\forall$I to a variable free in an undischarged assumption (so that, effectively, every assumption is always available for discharge).

Again, our strategy is to show that given one derivation, it is possible to construct another. In this case, we begin with an *AD* derivation (A) as below, with premises $\Delta \cup \{\mathscr{P}\}$. Treating $\mathscr{P}$ as an auxiliary premise, with scope as indicated in (B), we set out to show that there is an *AD* derivation (C), with premises in Δ alone, and lines numbered '1', '2', ... corresponding to 1, 2, ... in (A).

(F)

$$\begin{array}{lll}
\text{(A)}\ 1.\ \mathscr{Q}_1 & \text{(B)}\ 1.\ \mathscr{Q}_1 & \text{(C)}\ 1.\ \mathscr{P} \to \mathscr{Q}_1 \\
\quad\ \ 2.\ \mathscr{Q}_2 & \quad\ \ 2.\ \mathscr{Q}_2 & \quad\ \ 2.\ \mathscr{P} \to \mathscr{Q}_2 \\
\qquad\ \vdots & \qquad\ \vdots & \qquad\ \vdots \\
\qquad\ \mathscr{P} & \qquad\ \mid \mathscr{P} & \qquad\ \mathscr{P} \to \mathscr{P} \\
\qquad\ \vdots & \qquad\ \mid \vdots & \qquad\ \vdots \\
\quad\ \ n.\ \mathscr{Q}_n & \quad\ \ n.\ \mid \mathscr{Q}_n & \quad\ \ n.\ \mathscr{P} \to \mathscr{Q}_n
\end{array}$$

That is, we construct a derivation with premises in Δ such that for any formula $\mathscr{A}$ on line i of the first derivation, $\mathscr{P} \to \mathscr{A}$ appears on the line numbered 'i' of the constructed derivation. The last line n of the resultant derivation is the desired result, $\Delta \vdash_{AD} \mathscr{P} \to \mathscr{Q}$.

T9.3. (*Deduction Theorem*) If $\Delta \cup \{\mathscr{P}\} \vdash_{AD} \mathscr{Q}$, and no application of Gen under the scope of $\mathscr{P}$ is to a variable free in $\mathscr{P}$, then $\Delta \vdash_{AD} \mathscr{P} \to \mathscr{Q}$.

Suppose $A = \langle \mathscr{Q}_1, \mathscr{Q}_2, \ldots \mathscr{Q}_n \rangle$ is an *AD* derivation of $\mathscr{Q}$ from $\Delta \cup \{\mathscr{P}\}$, where $\mathscr{Q}$ is $\mathscr{Q}_n$ and no application of Gen under the scope of $\mathscr{P}$ is to a variable free in $\mathscr{P}$. By induction on the line numbers in derivation A, we show there

is a derivation C with premises only in Δ, such that for any line i of A, $\mathcal{P} \to \mathcal{Q}_i$ appears on the line numbered 'i' of C. The case when $i = n$ gives the desired result, that $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$.

Basis: $\mathcal{Q}_1$ of A is an axiom, a member of Δ, or $\mathcal{P}$ itself.

(i) If $\mathcal{Q}_1$ is an axiom or a member of Δ, then begin C as follows,

1.1	$\mathcal{Q}_1$	axiom / premise
1.2	$\mathcal{Q}_1 \to (\mathcal{P} \to \mathcal{Q}_1)$	A1
1	$\mathcal{P} \to \mathcal{Q}_1$	1.1, 1.2 MP

(ii) $\mathcal{Q}_1$ is $\mathcal{P}$ itself. By T3.1, $\vdash_{AD} \mathcal{P} \to \mathcal{P}$; which is to say $\mathcal{P} \to \mathcal{Q}_1$; so begin derivation C,

1	$\mathcal{P} \to \mathcal{P}$	T3.1

In either case, $\mathcal{P} \to \mathcal{Q}_1$ appears on the line numberd '1' of C with premises in Δ alone.

Assp: For any i, $1 \leq i < k$, $\mathcal{P} \to \mathcal{Q}_i$ appears on the line numbered 'i' of C, with premises in Δ alone.

Show: $\mathcal{P} \to \mathcal{Q}_k$ appears on the line numbered 'k' of C, with premises in Δ alone.

$\mathcal{Q}_k$ of A is a member of Δ, an axiom, $\mathcal{P}$ itself, or arises from previous lines by MP or Gen. If $\mathcal{Q}_k$ is a member of Δ, an axiom or $\mathcal{P}$ itself then, by reasoning as in the basis, $\mathcal{P} \to \mathcal{Q}_k$ appears on the line numbered 'k' of C from premises in Δ alone. So two cases remain.

(MP) If $\mathcal{Q}_k$ arises from previous lines by MP, then there are lines in derivation A of the sort,

i	$\mathcal{B}$	
	$\vdots$	
j	$\mathcal{B} \to \mathcal{C}$	
	$\vdots$	
k	$\mathcal{C}$	i,j MP

where i, j $< k$ and $\mathcal{Q}_k$ is $\mathcal{C}$. By assumption, there are lines in C,

i	$\mathcal{P} \to \mathcal{B}$
	$\vdots$
j	$\mathcal{P} \to (\mathcal{B} \to \mathcal{C})$

So continue derivation C as follows,

i	$\mathscr{P} \to \mathscr{B}$	
	$\vdots$	
j	$\mathscr{P} \to (\mathscr{B} \to \mathscr{C})$	
	$\vdots$	
k.1	$[\mathscr{P} \to (\mathscr{B} \to \mathscr{C})] \to [(\mathscr{P} \to \mathscr{B}) \to (\mathscr{P} \to \mathscr{C})]$	A2
k.2	$(\mathscr{P} \to \mathscr{B}) \to (\mathscr{P} \to \mathscr{C})$	j, k.1 MP
k	$\mathscr{P} \to \mathscr{C}$	i, k.2 MP

So $\mathscr{P} \to \mathscr{Q}_k$ appears on the line numbered 'k' of C, with premises in Δ alone.

(Gen) If $\mathscr{Q}_k$ arises from a previous line by Gen, then there are lines in derivation A of the sort,

i	$\mathscr{B} \to \mathscr{C}$
	$\vdots$
k	$\mathscr{B} \to \forall x \mathscr{C}$

where i < k, $\mathscr{Q}_k$ is $\mathscr{B} \to \forall x \mathscr{C}$ and x is not free in $\mathscr{B}$. Either line k is under the scope of $\mathscr{P}$ in derivation A or not.

(i) If line k is not under the scope of $\mathscr{P}$, then $\mathscr{B} \to \forall x \mathscr{C}$ in A follows from Δ alone. So continue C as follows to derive $\mathscr{B} \to \forall x \mathscr{C}$ and apply A1,

k.1	$\mathscr{Q}_1$	
k.2	$\mathscr{Q}_2$	exactly as in A but with prefix
	$\vdots$	'k.' for numeric references
k.k	$\mathscr{B} \to \forall x \mathscr{C}$	
k.k+1	$(\mathscr{B} \to \forall x \mathscr{C}) \to [\mathscr{P} \to (\mathscr{B} \to \forall x \mathscr{C})]$	A1
k	$\mathscr{P} \to (\mathscr{B} \to \forall x \mathscr{C})$	k.k+1, k.k MP

Since each of the lines in A up to k is derived from Δ alone, we have $\mathscr{P} \to \mathscr{Q}_k$ on the line numbered 'k' of C, from premises in Δ alone.

(ii) If line k is under the scope of $\mathscr{P}$, we depend on the assumption, and continue C as follows,

i	$\mathscr{P} \to (\mathscr{B} \to \mathscr{C})$	(by inductive assumption)
	$\vdots$	
k.1	$\mathscr{P} \to \forall x(\mathscr{B} \to \mathscr{C})$	i, Gen
k.2	$\forall x(\mathscr{B} \to \mathscr{C}) \to (\mathscr{B} \to \forall x \mathscr{C})$	T3.31
k	$\mathscr{P} \to (\mathscr{B} \to \forall x \mathscr{C})$	k.1, k.2 T3.2

If line k is under the scope of $\mathcal{P}$ then, since no application of Gen under the scope of $\mathcal{P}$ is to a variable free in $\mathcal{P}$, x is not free in $\mathcal{P}$; so k.1 meets the restriction on Gen. And since Gen is applied to line k in A, x is not free in $\mathcal{B}$; so line k.2 meets the restriction on T3.31. So we have $\mathcal{P} \to \mathcal{Q}_k$ on the line numbered 'k' of C, from premises in Δ alone.

Indct: For for any i, $\mathcal{P} \to \mathcal{Q}_k$ appears on the line numbered 'i' of C, from premises in Δ alone.

So given an *AD* derivation of $\mathcal{Q}$ from $\Delta \cup \{\mathcal{P}\}$, where no application of Gen under the scope of assumption $\mathcal{P}$ is to a variable free in $\mathcal{P}$, there is sure to be an *AD* derivation of $\mathcal{P} \to \mathcal{Q}$ from Δ alone. Notice that Gen*, T3.30 and T3.31 abbreviate sequences which include applications of Gen. So the restriction on Gen for the deduction theorem applies to applications of these results as well.

As a sample application of the deduction theorem (DT), let us consider another derivation of T3.2. In tis case, $\Delta = \{\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C}\}$, and we argue as follows,

(G)

1.	$\mathcal{A} \to \mathcal{B}$	prem
2.	$\mathcal{B} \to \mathcal{C}$	prem
3.	$\quad \mathcal{A}$	assp (g, DT)
4.	$\quad \mathcal{B}$	1,3 MP
5.	$\quad \mathcal{C}$	2,4 MP
6.	$\mathcal{A} \to \mathcal{C}$	3-5 DT

At line (5) we have established that $\Delta \cup \{\mathcal{A}\} \vdash_{AD} \mathcal{C}$; it follows from the deduction theorem that $\Delta \vdash_{AD} \mathcal{A} \to \mathcal{C}$. But we should be careful: this is not an *AD* derivation of $\mathcal{A} \to \mathcal{C}$ from $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{C}$. And it is not an abbreviation in the sense that we have seen so far — we do not appeal to a result whose derivation could be inserted at that very stage. Rather, what we have is a demonstration, via the deduction theorem, that there *exists* an *AD* derivation of $\mathcal{A} \to \mathcal{C}$ from the premises. If there is any abbreviating, the entire derivation abbreviates, or indicates the existence of, another. Our proof of the deduction theorem shows us that, given a derivation of $\Delta \cup \{\mathcal{P}\} \vdash_{AD} \mathcal{Q}$, it is possible to *construct* a derivation for $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$.

Let us see how this works in the example. Lines 1-5 become our derivation A, with $\Delta = \{\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C}\}$. For each $\mathcal{Q}_i$ in derivation A, the induction tells us how to derive $\mathcal{A} \to \mathcal{Q}_i$ from Δ alone. Thus $\mathcal{Q}_i$ on the first line is a member of Δ: reasoning from the basis tells us to use A1 as follows,

1.1	$\mathcal{A} \to \mathcal{B}$	prem
1.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to (\mathcal{A} \to \mathcal{B}))$	A1
1	$\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	1.2,1.1 MP

to get $\mathcal{A}$ arrow the form on line 1 of A. Notice that we are again using fractional line numbers to make lines in derivation A correspond to lines in the constructed derivation. One may wonder why we bother getting $\mathcal{A} \rightarrow \mathcal{Q}_1$. And again, the answer is that our "recipe" calls for this ingredient at stages connected to MP and Gen. Similarly, we can use A1 to get $\mathcal{A}$ arrow the form on line (2).

1.1	$\mathcal{A} \rightarrow \mathcal{B}$	prem
1.2	$(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B}))$	A1
1	$\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$	1.2,1.1 MP
2.1	$\mathcal{B} \rightarrow \mathcal{C}$	prem
2.2	$(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}))$	A1
2	$\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})$	2.2,2.1 MP

The form on line (3) is $\mathcal{A}$ itself. If we wanted a derivation in the primitive system, we could repeat the steps in our derivation of T3.1. But we will simply continue, as in the induction,

1.1	$\mathcal{A} \rightarrow \mathcal{B}$	prem
1.2	$(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B}))$	A1
1	$\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B}$	1.2,1.2 MP
2.1	$\mathcal{B} \rightarrow \mathcal{C})$	prem
2.2	$(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}))$	A1
2	$\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})$	2.2,2.1 MP
3	$\mathcal{A} \rightarrow \mathcal{A}$	T3.1

to get $\mathcal{A}$ arrow the form on line (3) of A. The form on line (4) arises from lines (1) and (3) by MP; reasoning in our show step tells us to continue,

1.1	$\mathcal{A} \rightarrow \mathcal{B}$	prem
1.2	$(\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B}))$	A1
1	$\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$	1.2,1.1 MP
2.1	$\mathcal{B} \rightarrow \mathcal{C}$	prem
2.2	$(\mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C}))$	A1
2	$\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})$	2.2,2.1 MP
3	$\mathcal{A} \rightarrow \mathcal{A}$	T3.1
4.1	$(\mathcal{A} \rightarrow (\mathcal{A} \rightarrow \mathcal{B})) \rightarrow ((\mathcal{A} \rightarrow \mathcal{A}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B}))$	A2
4.2	$(\mathcal{A} \rightarrow \mathcal{A}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B})$	4.1,1 MP
4	$\mathcal{A} \rightarrow \mathcal{B}$	4.2,3 MP

using A2 to get $\mathcal{A} \rightarrow \mathcal{B}$. Notice that the original justification from lines (1) and (3) dictates the appeal to (1) at line (4.2) and to (3) at line (4). The form on line (5) arises from lines (2) and (4) by MP; so, finally, we continue,

1.1	$\mathcal{A} \to \mathcal{B}$	prem
1.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to (\mathcal{A} \to \mathcal{B}))$	A1
1	$\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$	1.2,1.1 MP
2.1	$\mathcal{B} \to \mathcal{C}$	prem
2.2	$(\mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$	A1
2	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	2.2,2.1 MP
3	$\mathcal{A} \to \mathcal{A}$	T3.1
4.1	$(\mathcal{A} \to (\mathcal{A} \to \mathcal{B})) \to ((\mathcal{A} \to \mathcal{A}) \to (\mathcal{A} \to \mathcal{B}))$	A2
4.2	$(\mathcal{A} \to \mathcal{A}) \to (\mathcal{A} \to \mathcal{B})$	4.1,1 MP
4	$\mathcal{A} \to \mathcal{B}$	4.2,3 MP
5.1	$(\mathcal{A} \to (\mathcal{B} \to \mathcal{C})) \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C}))$	A2
5.2	$(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})$	5.1,2 MP
5	$\mathcal{A} \to \mathcal{C}$	5.2,4 MP

And we have the *AD* derivation which our proof of the deduction theorem told us there would be. Notice that this derivation is not very efficient! We did it in seven lines (without appeal to T3.1) in chapter 3. What our proof of the deduction theorem tells us is that there is sure to be some derivation — where there is no expectation that the guaranteed derivation is particularly elegant or efficient.

Here is a last example which makes use of the deduction theorem. First, an alternate derivation of T3.3.

(H)

1.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	prem
2.	$\quad\vert\ \mathcal{B}$	assp (g, DT)
3.	$\quad\vert\quad\vert\ \mathcal{A}$	assp (g, DT)
4.	$\quad\vert\quad\vert\ \mathcal{B} \to \mathcal{C}$	1,3 MP
5.	$\quad\vert\quad\vert\ \mathcal{C}$	4,2 MP
6.	$\quad\vert\ \mathcal{A} \to \mathcal{C}$	3-5 DT
7.	$\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$	2-6 DT

In chapter 3 we proved T3.3 in five lines (with an appeal to T3.2). But perhaps this version is relatively intuitive, coinciding as it does, with strategies from *ND*. In this case, there are two applications of DT, and reasoning from the induction therefore applies twice. First, at line (5), there is an *AD* derivation of $\mathcal{C}$ from $\{\mathcal{A} \to (\mathcal{B} \to \mathcal{C}), \mathcal{B}\} \cup \{\mathcal{A}\}$. By reasoning from the induction, then, there is an *AD* derivation from just $\{\mathcal{A} \to (\mathcal{B} \to \mathcal{C}), \mathcal{B}\}$ with $\mathcal{A}$ arrow each of the forms on lines 1-5. So there is a derivation of $\mathcal{A} \to \mathcal{C}$ from $\{\mathcal{A} \to (\mathcal{B} \to \mathcal{C}), \mathcal{B}\}$. But then reasoning from the induction applies again. By reasoning from the induction applied to this *new* derivation, there is a derivation from just $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$ with $\mathcal{B}$ arrow each of the forms in it. So there is a derivation of $\mathcal{B} \to (\mathcal{A} \to \mathcal{C})$ from just $\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$. So the first derivation, lines 1-5 above, is replaced by another, by the reasoning from

DT. Then *it* is replaced by another, again given the reasoning from DT. The result is an *AD* derivation of the desired result.

Here are a couple more cases, where the latter at least, may inspire a certain affection for the deduction theorem.

T9.4. $\vdash_{AD} \mathcal{A} \to (\mathcal{B} \to (\mathcal{A} \wedge \mathcal{B}))$

T9.5. $\vdash_{AD} (\mathcal{A} \to \mathcal{C}) \to [(\mathcal{B} \to \mathcal{C}) \to ((\mathcal{A} \vee \mathcal{B}) \to \mathcal{C})]$

E9.5. Making use of the deduction theorem, prove T9.4 and T9.5. Having done so, see if you can prove them in the style of chapter 3, without any appeal to DT.

E9.6. By the method of our proof of the deduction theorem, convert the above derivation (H) for T3.3 into an official *AD* derivation. Hint: As described above, the method of the induction applies twice: first to lines 1-5, and then to the new derivation. The result should be derivations with 13, and then 37 lines.

E9.7. Consider the axiomatic system *A2* from E3.4 on p. 79, and produce a demonstration of the deduction theorem for it. That is, show that if $\Delta \cup \{\mathcal{P}\} \vdash_{A2} \mathcal{Q}$, then $\Delta \vdash_{A2} \mathcal{P} \to \mathcal{Q}$. You may appeal to any of the *A2* theorems listed on 79.

9.3.2 Substitution Theorems

Recall what we are after. Our goal is to show that if $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. Toward this end, the deduction theorem lets AD mimic rules in ND which require subderivations. For equality, we turn to some substitution results. Say a complex term r is *free* in an expression $\mathcal{P}$ just in case no variable in r is bound. Then where $\mathcal{T}$ is any term or formula, let $\mathcal{T}^{r}/\!\!/_{s}$ be $\mathcal{T}$ where at most one free instance of r is replaced by term s. Having shown in T3.37, that $\vdash_{AD} (q_i = s) \to (\mathcal{R}^n q_1 \ldots q_i \ldots q_n \to \mathcal{R}^n q_1 \ldots s \ldots q_n)$, one might think we have proved that $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}^{r}/\!\!/_{s})$ for any atomic formula $\mathcal{A}$ and any terms r and s. But *this is not so*. Similarly, having proved in T3.36 that $\vdash_{AD} (q_i = s) \to (h^n q_1 \ldots q_i \ldots q_n = h^n q_1 \ldots s \ldots q_n)$, one might think we have proved that $\vdash_{AD} (r = s) \to (t \to t^{r}/\!\!/_{s})$ for any terms r, s and t. But this is not so. In each case, the difficulty is that the replaced term r might be a *component* of the other terms $q_1 \ldots q_n$, and so might not be any of

$\mathscr{q}_1 \dots \mathscr{q}_n$. What we have shown is only that it is possible to replace any of the whole terms, $\mathscr{q}_1 \dots \mathscr{q}_n$. Thus, $(x = y) \rightarrow (f^1g^1x = f^1g^1y)$ is not an instance of T3.36 because we do not replace g^1x but rather a component of it.

However, as one might expect, it is possible to replace terms in basic parts; use the result to make replacements in terms of which *they* are parts; and so forth, all the way up to wholes. Both $(x = y) \rightarrow (g^1x = g^1y)$ and $(g^1x = g^1y) \rightarrow (f^1g^1x = f^1g^1y)$ are instances of T3.36. (Be clear about these examples in your mind.) From these, with T3.2 it follows that $(x = y) \rightarrow (f^1g^1x = f^1g^1y)$. This example suggests a method for obtaining the more general results: Using T3.36, we work from equalities at the level of the parts, to equalities at the level of the whole. For the case of terms, the proof is by induction on the number of function symbols in an arbitrary term t.

T9.6. For arbitrary terms $\mathscr{r}$, $\mathscr{s}$ and t, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Basis: If t has no function symbols, then t is a variable or a constant. In this case, either (i) $\mathscr{r} \neq t$ and $t^{\mathscr{r}}/\!\!/_{\mathscr{s}} = t$ (nothing is replaced) or (ii) $\mathscr{r} = t$ and $t^{\mathscr{r}}/\!\!/_{\mathscr{s}} = \mathscr{s}$ (all of t is replaced). (i) In this case, by T3.32, $\vdash_{AD}\ t = t$; which is to say, $\vdash_{AD} (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$; so with A1, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$. (ii) In this case, $(\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$ is the same as $(\mathscr{r} = \mathscr{s}) \rightarrow (\mathscr{r} = \mathscr{s})$; so by T3.1, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Show: If t has k function symbols, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

If t has k function symbols, then t is of the form $\mathscr{h}^n\mathscr{q}_1 \dots \mathscr{q}_n$ for terms $\mathscr{q}_1 \dots \mathscr{q}_n$ with $< k$ function symbols. If all of t is replaced, or no part of t is replaced, then reason as in the basis. So suppose $\mathscr{r}$ is some subcomponent of t; then for some $\mathscr{q}_i$, $t^{\mathscr{r}}/\!\!/_{\mathscr{s}}$ is $\mathscr{h}^n\mathscr{q}_1 \dots \mathscr{q}_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \dots \mathscr{q}_n$. By assumption, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (\mathscr{q}_i = \mathscr{q}_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$; and by T3.36, $\vdash_{AD} (\mathscr{q}_i = \mathscr{q}_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}}) \rightarrow (\mathscr{h}^n\mathscr{q}_1 \dots \mathscr{q}_i \dots \mathscr{q}_n = \mathscr{h}^n\mathscr{q}_1 \dots \mathscr{q}_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \dots \mathscr{q}_n)$; so by T3.2, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (\mathscr{h}^n\mathscr{q}_1 \dots \mathscr{q}_i \dots \mathscr{q}_n = \mathscr{h}^n\mathscr{q}_1 \dots \mathscr{q}_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \dots \mathscr{q}_n)$; but this is to say, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Indct: For any terms $\mathscr{r}$, $\mathscr{s}$ and t, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \rightarrow (t = t^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

We might think of this result as a further strengthened or generalized version of the *AD* axiom A6. Where A6 lets us replace just variables in terms of the sort $\mathscr{h}^n x_1 \dots x_n$, we are now in a position to replace in arbitrary terms with arbitrary terms.

Now we can go after a similarly strengthened version of A7. We show that for any formula $\mathcal{A}$, if $\mathscr{s}$ is free for the replaced instance of $\mathscr{r}$ in $\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$. The argument is by induction on the number of operators in $\mathcal{A}$.

T9.7. For any formula $\mathcal{A}$ and terms $\mathscr{r}$ and $\mathscr{s}$, if $\mathscr{s}$ is free for the replaced instance of $\mathscr{r}$ in $\mathcal{A}$, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Consider an arbitrary $\mathscr{r}$, $\mathscr{s}$ and $\mathcal{A}$, and suppose $\mathscr{s}$ is free for the replaced instance of $\mathscr{r}$ in $\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$.

Basis: If $\mathcal{A}$ is atomic then (i) $\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} = \mathcal{A}$ (nothing is replaced) or (ii) $\mathcal{A}$ is an atomic of the form $\mathcal{R}^n t_1 \dots t_i \dots t_n$ and $\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$ is $\mathcal{R}^n t_1 \dots t_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \dots t_n$. (i) In this case, by T3.1, $\vdash_{AD} \mathcal{A} \to \mathcal{A}$, which is to say $\vdash_{AD} \mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$; so with A1, $\vdash_{AD} \mathscr{r} = \mathscr{s} \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$. (ii) In this case, by T9.6, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (t_i = t_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$; and by T3.37, $\vdash_{AD} (t_i = t_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}}) \to (\mathcal{R}^n t_1 \dots t_i \dots t_n \to \mathcal{R}^n t_1 \dots t_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \dots t_n)$; so by T3.2, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{R}^n t_1 \dots t_i \dots t_n \to \mathcal{R}^n t_1 \dots t_i{}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \dots t_n)$; and this is just to say, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Assp: For any i, $0 \le i < k$, if $\mathcal{A}$ has i operator symbols and $\mathscr{s}$ is free for the replaced instance of $\mathscr{r}$ in $\mathcal{A}$, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

Corollary to the assumption. If $\mathcal{A}$ has $< k$ operators, then $\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$ has $< k$ operators; and since $\mathscr{s}$ replaces only a free instance of $\mathscr{r}$ in $\mathcal{A}$, $\mathscr{r}$ is free for the replacing instance of $\mathscr{s}$ in $\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$; so where the outer substitution is made to sustain $[\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}]^{\mathscr{s}}/\!\!/_{\mathscr{r}} = \mathcal{A}$, we have $\vdash_{AD} (\mathscr{s} = \mathscr{r}) \to (\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to [\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}]^{\mathscr{s}}/\!\!/_{\mathscr{r}})$ as an instance of the inductive assumption, which is just, $\vdash_{AD} (\mathscr{s} = \mathscr{r}) \to (\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{A})$. And by T3.33, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathscr{s} = \mathscr{r})$; so with T3.2, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{A})$.

Show: If $\mathcal{A}$ has k operator symbols and $\mathscr{s}$ is free for the replaced instance of $\mathscr{r}$ in $\mathcal{A}$, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

If $\mathcal{A}$ has k operator symbols, then $\mathcal{A}$ is of the form, $\sim\mathcal{P}$, $\mathcal{P} \to \mathcal{Q}$ or $\forall x \mathcal{P}$ for variable x and formulas $\mathcal{P}$ and $\mathcal{Q}$ with $< k$ operator symbols. Suppose $\mathscr{s}$ is free for any replaced instance of $\mathscr{r}$ in $\mathcal{A}$.

($\sim$) Suppose $\mathcal{A}$ is $\sim\mathcal{P}$. Then $\mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}}$ is $[\sim\mathcal{P}]^{\mathscr{r}}/\!\!/_{\mathscr{s}}$ which is the same as $\sim[\mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}}]$. Since $\mathscr{s}$ is free for a replaced instance of $\mathscr{r}$ in $\mathcal{A}$, it is free for that instance of $\mathscr{r}$ in $\mathcal{P}$; so by the corollary to the assumption, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{P})$. But by T3.13, $\vdash_{AD} (\mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}} \to \mathcal{P}) \to (\sim\mathcal{P} \to \sim[\mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}}])$; so by T3.2, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\sim\mathcal{P} \to \sim[\mathcal{P}^{\mathscr{r}}/\!\!/_{\mathscr{s}}])$; which is to say, $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

($\rightarrow$) Suppose $\mathcal{A}$ is $\mathcal{P} \rightarrow \mathcal{Q}$. Then $\mathcal{A}^{r}/\!\!/_{s}$ is $\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{Q}$ or $\mathcal{P} \rightarrow \mathcal{Q}^{r}/\!\!/_{s}$. (i) In the former case, since s is free for a replaced instance of r in $\mathcal{A}$, it is free for that instance of r in $\mathcal{P}$; so by the corollary to the assumption, $\vdash_{AD} (r = s) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{P})$; so we may reason as follows,

1.	$(r = s) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{P})$	prem
2.	$\mid\ r = s$	assp (g, DT)
3.	$\mid\ \mid\ \mathcal{P} \rightarrow \mathcal{Q}$	assp (g, DT)
4.	$\mid\ \mid\ \mid\ \mathcal{P}^{r}/\!\!/_{s}$	assp (g, DT)
5.	$\mid\ \mid\ \mid\ \mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{P}$	1,2 MP
6.	$\mid\ \mid\ \mid\ \mathcal{P}$	5,4 MP
7.	$\mid\ \mid\ \mid\ \mathcal{Q}$	3,6 MP
8.	$\mid\ \mid\ \mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{Q}$	4-7 DT
9.	$\mid\ (\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{Q})$	3-8 DT
10.	$(r = s) \rightarrow [(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{Q})]$	2-9 DT

So $\vdash_{AD} (r = s) \rightarrow [(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P}^{r}/\!\!/_{s} \rightarrow \mathcal{Q})]$; which is to say, $\vdash_{AD} (r = s) \rightarrow (\mathcal{A} \rightarrow \mathcal{A}^{r}/\!\!/_{s})$. (ii) And similarly in the other case [by homework], $\vdash_{AD} (r = s) \rightarrow [(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\mathcal{P} \rightarrow \mathcal{Q}^{r}/\!\!/_{s})]$. So in either case, $\vdash_{AD} (r = s) \rightarrow (\mathcal{A} \rightarrow \mathcal{A}^{r}/\!\!/_{s})$.

($\forall$) Suppose $\mathcal{A}$ is $\forall x\mathcal{P}$. Then a free instance of r in $\mathcal{A}$ remains free in $\mathcal{P}$ and $\mathcal{A}^{r}/\!\!/_{s}$ is $\forall x[\mathcal{P}^{r}/\!\!/_{s}]$. Since s is free for r in $\mathcal{A}$, s is free for r in $\mathcal{P}$; so by assumption, $\vdash_{AD} (r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$; so we may reason as follows,

1.	$(r = s) \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s})$	prem
2.	$\mid\ r = s$	assp (g, DT)
3.	$\mid\ \forall x\mathcal{P} \rightarrow \mathcal{P}$	A4
4.	$\mid\ \mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s}$	1,2 MP
5.	$\mid\ \forall x\mathcal{P} \rightarrow \mathcal{P}^{r}/\!\!/_{s}$	3,4 T3.2
6.	$\mid\ \forall x\mathcal{P} \rightarrow \forall x\mathcal{P}^{r}/\!\!/_{s}$	5 Gen
7.	$(r = s) \rightarrow (\forall x\mathcal{P} \rightarrow \forall x\mathcal{P}^{r}/\!\!/_{s})$	2-6 DT

Notice that x is sure to be free for itself in $\mathcal{P}$, so that (3) is an instance of A4. And x is bound in $\forall x\mathcal{P}$, so (6) is an instance of Gen. And because r is free in $\mathcal{A}$, and s is free for r in $\mathcal{A}$, x cannot be a variable in r or s; so the restriction on DT is met at (7). So $\vdash_{AD} (r = s) \rightarrow (\forall x\mathcal{P} \rightarrow \forall x\mathcal{P}^{r}/\!\!/_{s})$; which is to say, $\vdash_{AD} (r = s) \rightarrow (\mathcal{A} \rightarrow \mathcal{A}^{r}/\!\!/_{s})$.

So for any $\mathcal{A}$ with k operator symbols, $\vdash_{AD} (r = s) \rightarrow (\mathcal{A} \rightarrow \mathcal{A}^{r}/\!\!/_{s})$.

Indct: For any $\mathcal{A}$, $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}^{r}/\!\!/_{s})$.

So T9.7, for any formula $\mathcal{A}$, and terms r and s, if s is free for a replaced instance of r in $\mathcal{A}$, then $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}^{r}/\!\!/_{s})$.

It is a short step from T9.7, which allows substitution of just a single term, to T9.8 which allows substitution of arbitrarily many. Where, as in chapter 6, $\mathcal{P}^{t}/_{s}$ is $\mathcal{P}$ with some, but not necessarily all, free instances of term t replaced by term s,

T9.8. For any formula $\mathcal{A}$ and terms r and s, if s is free for the replaced instances of r in $\mathcal{A}$, then $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}^{r}/_{s})$.

By induction on the number of instances of r that are replaced by s in $\mathcal{A}$. Say $\mathcal{A}_i$ is $\mathcal{A}$ with i free instances of r replaced by s. Suppose s is free for the replaced instances of r in $\mathcal{A}$. We show that for any i, $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}_i)$.

Basis: If no instances of r are replaced by s then $\mathcal{A}_0 = \mathcal{A}$. But by T3.1, $\vdash_{AD} \mathcal{A} \to \mathcal{A}$, and by A1, $\vdash_{AD} (\mathcal{A} \to \mathcal{A}) \to [(r = s) \to (\mathcal{A} \to \mathcal{A})]$; so by MP, $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A})$; which is to say, $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}_0)$.

Assp: For any i, $0 \le i < k$, $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}_i)$.

Show: $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}_k)$.

$\mathcal{A}_k$ is of the sort $\mathcal{A}_i{}^{r}/\!\!/_{s}$ for $i < k$. By assumption, then, $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}_i)$, and by T9.7, $\vdash_{AD} (r = s) \to (\mathcal{A}_i \to \mathcal{A}_i{}^{r}/\!\!/_{s})$, which is the same as $\vdash_{AD} (r = s) \to (\mathcal{A}_i \to \mathcal{A}_k)$. So reason as follows,

1.	$(r = s) \to (\mathcal{A} \to \mathcal{A}_i)$	by assumption
2.	$(r = s) \to (\mathcal{A}_i \to \mathcal{A}_k)$	T9.7
3.	$\quad\mid r = s$	assp (g, DT)
4.	$\quad\mid \mathcal{A} \to \mathcal{A}_i$	1,3 MP
5.	$\quad\mid \mathcal{A}_i \to \mathcal{A}_k$	2,3 MP
6.	$\quad\mid \mathcal{A} \to \mathcal{A}_k$	4,5 T3.2
7.	$(r = s) \to (\mathcal{A} \to \mathcal{A}_k)$	3-6 DT

Since s is free for the replaced instances of r in $\mathcal{A}$, (2) is an instance of T9.7. So $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}_k)$.

Indct: For any i, $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}_i)$.

In effect, the result is by multiple applications of T9.7. No matter how many instances of $\mathscr{r}$ have been replaced by $\mathscr{s}$, we may use T9.7 to replace another!

A final substitution result allows substitution of *formulas* rather than terms. Where $\mathcal{A}^{\mathcal{B}}/\!\!/_{\mathcal{C}}$ is $\mathcal{A}$ with exactly one instance of a subformula $\mathcal{B}$ replaced by formula $\mathcal{C}$,

T9.9. For any formulas $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, if $\vdash_{AD} \mathcal{B} \leftrightarrow \mathcal{C}$, then $\vdash_{AD} \mathcal{A} \leftrightarrow \mathcal{A}^{\mathcal{B}}/\!\!/_{\mathcal{C}}$.

The proof is by induction on the number of operators in $\mathcal{A}$. If you have understood the previous two inductions, this one should be straightforward. Observe that, in the basis, when $\mathcal{A}$ is atomic, $\mathcal{B}$ can only be all of $\mathcal{A}$, and $\mathcal{A}^{\mathcal{B}}/\!\!/_{\mathcal{C}}$ is $\mathcal{C}$. For the show, either $\mathcal{B}$ is all of $\mathcal{A}$ or it is not. If it is, then the result holds by reasoning as in the basis. If $\mathcal{B}$ is a proper part of $\mathcal{A}$, then the assumption applies.

*E9.8. Set up the above demonstration for T9.7 and complete the unfinished case to provide a complete demonstration that for any formula $\mathcal{A}$, and terms $\mathscr{r}$ and $\mathscr{s}$, if $\mathscr{s}$ is free for the replaced instance of $\mathscr{r}$ in $\mathcal{A}$, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$.

E9.9. Suppose our primitive operators are $\sim$, $\wedge$ and $\exists$ rather than $\sim$, $\to$ and $\forall$. Modify your argument for T9.7 to show that for any formula $\mathcal{A}$, and terms $\mathscr{r}$ and $\mathscr{s}$, if $\mathscr{s}$ is free for the replaced instance of $\mathscr{r}$ in $\mathcal{A}$, then $\vdash_{AD} (\mathscr{r} = \mathscr{s}) \to (\mathcal{A} \to \mathcal{A}^{\mathscr{r}}/\!\!/_{\mathscr{s}})$. Hint: Do not forget that you may appeal to T9.4.

*E9.10. Prove T9.9, to show that for any formulas $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, if $\vdash_{AD} \mathcal{B} \leftrightarrow \mathcal{C}$, then $\vdash_{AD} \mathcal{A} \leftrightarrow \mathcal{A}^{\mathcal{B}}/\!\!/_{\mathcal{C}}$. Hint: Where $\mathcal{P} \leftrightarrow \mathcal{Q}$ abbreviates $(\mathcal{P} \to \mathcal{Q}) \wedge (\mathcal{Q} \to \mathcal{P})$, you can use (abv) along with T3.19, T3.20 and T9.4 to manipulate formulas of the sort $\mathcal{P} \leftrightarrow \mathcal{Q}$.

E9.11. Where $\mathcal{A}^{\mathcal{B}}/_{\mathcal{C}}$ replaces some, but not necessarily all, instances of formula $\mathcal{B}$ with formula $\mathcal{C}$, use your result from E9.10 to show that if $\vdash_{AD} \mathcal{B} \leftrightarrow \mathcal{C}$, then $\vdash_{AD} \mathcal{A} \leftrightarrow \mathcal{A}^{\mathcal{B}}/_{\mathcal{C}}$.

9.3.3 Intended Result

We are finally ready to show that if $\Gamma \vdash_{ND} \mathcal{P}$ then $\Gamma \vdash_{AD} \mathcal{P}$. As usual, the idea is that the existence of one derivation guarantees the existence of another. In this case, we begin with a derivation in *ND*, and move to the existence of one in *AD*. Suppose $\Gamma \vdash_{ND} \mathcal{P}$. Then there is an *ND* derivation N of $\mathcal{P}$ from premises in Γ, with lines $\langle \mathcal{Q}_1 \ldots \mathcal{Q}_n \rangle$ and $\mathcal{Q}_n = \mathcal{P}$. We show that there is an *AD* derivation *A* of the same result (with possible appeal to DT). Say derivation *A matches N* iff any $\mathcal{Q}_i$ from *N* appears at the same scope on the line numbered 'i' of A; and say derivation A is *good* iff it has no application of Gen to a variable free in an undischarged auxiliary assumption. Then, given derivation N, we show that there is a good derivation A that matches N. The reason for the restriction on free variables is to be sure that DT is available at any stage in derivation A. The argument is by induction on the line number of N, where we show that for any i, there is a good derivation A_i that matches N through line i. The case when $i = n$ is an *AD* derivation of $\mathcal{P}$ under the scope of the premises alone, and so a demonstration of the desired result.

T9.10. If $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

Suppose $\Gamma \vdash_{ND} \mathcal{P}$; then there is an *ND* derivation N of $\mathcal{P}$ from premises in Γ. We show that for any i, there is a good *AD* derivation A_i that matches N through line i.

Basis: The first line of N is a premise or an assumption. Let A_1 be the same. Then A_1 matches N; and since there is no application of Gen, A_1 is good.

Assp: For any i, $1 \leq i < k$, there is a good derivation A_i that matches N through line i.

Show: There is a good derivation A_k that matches N through line k.

Either $\mathcal{Q}_k$ is a premise or assumption, or arises from previous lines by R, $\wedge$E, $\wedge$I, $\rightarrow$E, $\rightarrow$I, $\sim$E, $\sim$I, $\vee$E, $\vee$I, $\leftrightarrow$E, $\leftrightarrow$I, $\forall$E, $\forall$I, $\exists$E, $\exists$I, =E or =I.

(p/a) If $\mathcal{Q}_k$ is a premise or an assumption, let A_k continue in the same way. Then, by reasoning as in the basis, A_k matches N and is good.

(R) If $\mathcal{Q}_k$ arises from previous lines by R, then N looks something like this,

$$\begin{array}{r|ll} i & \mathcal{B} & \\ & & \\ k & \mathcal{B} & \quad i \text{ R} \end{array}$$

where $i < k$, $\mathcal{B}$ is accessible at line k, and $\mathcal{Q}_k = \mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}$ appears at the same scope on the line numbered 'i' of A_{k-1} and is accessible in A_{k-1}. So let A_k continue as follows,

$$\begin{array}{r|ll} i & \mathcal{B} & \\ & \vdots & \\ k.1 & \mathcal{B} \to \mathcal{B} & \text{T3.1} \\ k & \mathcal{B} & k.1,i \text{ MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

(∧E) If $\mathcal{Q}_k$ arises by ∧E, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} \wedge \mathcal{C} & \\ & & \\ k & \mathcal{B} & i \wedge\text{E} \end{array} \qquad \text{or} \qquad \begin{array}{r|ll} i & \mathcal{B} \wedge \mathcal{C} & \\ & & \\ k & \mathcal{C} & i \wedge\text{E} \end{array}$$

where $i < k$ and $\mathcal{B} \wedge \mathcal{C}$ is accessible at line k. In the first case, $\mathcal{Q}_k = \mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B} \wedge \mathcal{C}$ appears at the same scope on the line numbered 'i' of A_{k-1} and is accessible in A_{k-1}. So let A_k continue as follows,

$$\begin{array}{r|ll} i & \mathcal{B} \wedge \mathcal{C} & \\ & & \\ k.1 & (\mathcal{B} \wedge \mathcal{C}) \to \mathcal{B} & \text{T3.20} \\ k & \mathcal{B} & k.1,i \text{ MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good. And similarly in the other case, by application of T3.19.

(∧I) If $\mathcal{Q}_k$ arises from previous lines by ∧I, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} & \\ j & \mathcal{C} & \\ & & \\ k & \mathcal{B} \wedge \mathcal{C} & i,j \wedge\text{I} \end{array}$$

where $i, j < k$, $\mathcal{B}$ and $\mathcal{C}$ are accessible at line k, and $\mathcal{Q}_k = \mathcal{B} \wedge \mathcal{C}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}$ and $\mathcal{C}$ appear at the same scope on the lines numbered 'i' and 'j' of A_{k-1} and are accessible in A_{k-1}. So let A_k continue as follows,

$$\begin{array}{r|ll} i & \mathcal{B} & \\ j & \mathcal{C} & \\ & & \\ k.1 & \mathcal{B} \rightarrow (\mathcal{C} \rightarrow (\mathcal{B} \wedge \mathcal{C})) & \text{T9.4} \\ k.2 & \mathcal{C} \rightarrow (\mathcal{B} \wedge \mathcal{C}) & k.1,i \text{ MP} \\ k & \mathcal{B} \wedge \mathcal{C} & k.2,j \text{ MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

(→E) If $\mathcal{Q}_k$ arises from previous lines by →E, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} \rightarrow \mathcal{C} & \\ j & \mathcal{B} & \\ & & \\ k & \mathcal{C} & i,j \rightarrow\text{E} \end{array}$$

where $i, j < k$, $\mathcal{B} \rightarrow \mathcal{C}$ and $\mathcal{B}$ are accessible at line k, and $\mathcal{Q}_k = \mathcal{C}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B} \rightarrow \mathcal{C}$ and $\mathcal{B}$ appear at the same scope on the lines numbered 'i' and 'j' of A_{k-1} and are accessible in A_{k-1}. So let A_k continue as follows,

$$\begin{array}{r|ll} i & \mathcal{B} \rightarrow \mathcal{C} & \\ j & \mathcal{B} & \\ & & \\ k & \mathcal{C} & i,j \text{ MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

(→I) If $\mathcal{Q}_k$ arises by →I, then N is something like this,

$$\begin{array}{r|l|ll} i & & \underline{\mathcal{B}} & \\ & & & \\ j & & \mathcal{C} & \\ k & \multicolumn{2}{l}{\mathcal{B} \rightarrow \mathcal{C}} & i\text{-}j \rightarrow\text{I} \end{array}$$

where $i, j < k$, the subderivation is accessible at line k and $\mathcal{Q}_k = \mathcal{B} \rightarrow \mathcal{C}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}$ and $\mathcal{C}$ appear at the same scope on the lines numbered 'i' and 'j' of A_{k-1}; since they appear at the same scope, the parallel

subderivation is accessible in A_{k-1}; since A_{k-1} is good, no application of Gen under the scope of $\mathcal{B}$ is to a variable free in $\mathcal{B}$. So let A_k continue as follows,

$$\begin{array}{r|ll}
i & \quad \mathcal{B} & \\
 & \quad \vdots & \\
j & \quad \mathcal{C} & \\
k & \mathcal{B} \to \mathcal{C} & i\text{-}j \text{ DT}
\end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\sim$E) If $\mathcal{Q}_k$ arises by $\sim$E , then N is something like this (reverting to the unabbreviated form),

$$\begin{array}{r|ll}
i & \quad \sim\mathcal{B} & \\
 & \quad \vdots & \\
j & \quad \mathcal{C} \wedge \sim\mathcal{C} & \\
k & \mathcal{B} & i\text{-}j \ \sim\text{E}
\end{array}$$

where $i, j < k$, the subderivation is accessible at line k, and $\mathcal{Q}_k = \mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\sim\mathcal{B}$ and $\mathcal{C} \wedge \sim\mathcal{C}$ appear at the same scope on the lines numbered 'i' and 'j' of A_{k-1}; since they appear at the same scope, the parallel subderivation is accessible in A_{k-1}; since A_{k-1} is good, no application of Gen under the scope of $\sim\mathcal{B}$ is to a variable free in $\sim\mathcal{B}$. So let A_k continue as follows,

$$\begin{array}{r|ll}
i & \quad \sim\mathcal{B} & \\
 & \quad \vdots & \\
j & \quad \mathcal{C} \wedge \sim\mathcal{C} & \\
k.1 & \sim\mathcal{B} \to (\mathcal{C} \wedge \sim\mathcal{C}) & i\text{-}j \text{ DT} \\
k.2 & (\mathcal{C} \wedge \sim\mathcal{C}) \to \mathcal{C} & \text{T3.20} \\
k.3 & (\mathcal{C} \wedge \sim\mathcal{C}) \to \sim\mathcal{C} & \text{T3.19} \\
k.4 & \sim\mathcal{B} \to \mathcal{C} & k.1, k.2 \text{ T3.2} \\
k.5 & \sim\mathcal{B} \to \sim\mathcal{C} & k.1, k.3 \text{ T3.2} \\
k.6 & (\sim\mathcal{B} \to \sim\mathcal{C}) \to ((\sim\mathcal{B} \to \mathcal{C}) \to \mathcal{B}) & \text{A3} \\
k.7 & (\sim\mathcal{B} \to \mathcal{C}) \to \mathcal{B} & k.6, k.5 \text{ MP} \\
k & \mathcal{B} & k.7, k.4 \text{ MP}
\end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\sim$I) Homework.

($\vee$E) If $\mathcal{Q}_k$ arises by $\vee$E, then N is something like this,

$$
\begin{array}{r|l l}
f & \mathcal{B} \vee \mathcal{C} & \\
g & \quad \mathcal{B} & \\
h & \quad \mathcal{D} & \\
i & \quad \mathcal{C} & \\
j & \quad \mathcal{D} & \\
k & \mathcal{D} & f,g\text{-}h,i\text{-}j\ \vee\text{E}
\end{array}
$$

where $f, g, h, i, j < k$, $\mathcal{B} \vee \mathcal{C}$ and the two subderivations are accessible at line k and $\mathcal{Q}_k = \mathcal{D}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So the formulas at lines f, g, h, i, j appear at the same scope on corresponding lines in A_{k-1}; since they appear at the same scope, $\mathcal{B} \vee \mathcal{C}$ and corresponding subderivations are accessible in A_{k-1}; since A_{k-1} is good, no application of Gen under the scope of $\mathcal{B}$ is to a variable free in $\mathcal{B}$, and no application of Gen under the scope of $\mathcal{C}$ is to a variable free in $\mathcal{C}$. So let A_k continue as follows,

$$
\begin{array}{r|l l}
f & \mathcal{B} \vee \mathcal{C} & \\
g & \quad \mathcal{B} & \\
h & \quad \mathcal{D} & \\
i & \quad \mathcal{C} & \\
j & \quad \mathcal{D} & \\
k.1 & \mathcal{B} \rightarrow \mathcal{D} & g\text{-}h \text{ DT} \\
k.2 & \mathcal{C} \rightarrow \mathcal{D} & i\text{-}j \text{ DT} \\
k.3 & (\mathcal{B} \rightarrow \mathcal{D}) \rightarrow [(\mathcal{C} \rightarrow \mathcal{D}) \rightarrow ((\mathcal{B} \vee \mathcal{C}) \rightarrow \mathcal{D})] & \text{T9.5} \\
k.4 & (\mathcal{C} \rightarrow \mathcal{D}) \rightarrow ((\mathcal{B} \vee \mathcal{C}) \rightarrow \mathcal{D}) & k.3,k.1 \text{ MP} \\
k.5 & (\mathcal{B} \vee \mathcal{C}) \rightarrow \mathcal{D} & k.4,k.2 \text{ MP} \\
k & \mathcal{D} & k.5,f \text{ MP}
\end{array}
$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

($\vee$I) Homework.

($\leftrightarrow$E) Homework.

($\leftrightarrow$I) Homework.

($\forall$E) Homework.

($\forall$I) If $\mathcal{Q}_k$ arises by $\forall$I, then N looks something like this,

$$\begin{array}{r|ll} i & \mathcal{B}^x_v & \\ & & \\ k & \forall x\mathcal{B} & i\ \forall\text{I} \end{array}$$

where $i < k$, $\mathcal{B}^x_v$ is accessible at line k, and $\mathcal{Q}_k = \forall x\mathcal{B}$; further the *ND* restrictions on $\forall$I are met: (i) v is free for x in $\mathcal{B}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\forall x\mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B}^x_v$ appears at the same scope on the line numbered 'i' of A_{k-1} and is accessible in A_{k-1}. So let A_k continue as follows,

$$\begin{array}{r|ll} i & \mathcal{B}^x_v & \\ & & \\ k.1 & \forall v\mathcal{B}^x_v & i\ \text{Gen*} \\ k.2 & \forall v\mathcal{B}^x_v \to \forall x\mathcal{B} & \text{T3.27} \\ k & \forall x\mathcal{B} & k.1,k.2\ \text{MP} \end{array}$$

If v is x, we have the desired result already at $k.1$. So suppose $x \neq v$. On its face, $k.2$ does not look like T3.27 according to which $\forall x\mathcal{A} \to \forall y\mathcal{A}^x_y$ with y free for x in $\mathcal{A}$ but not free in $\forall x\mathcal{A}$. To see that we have it right, consider first, $\forall v\mathcal{B}^x_v \to \forall x[\mathcal{B}^x_v]^v_x$; this is an instance of T3.27 so long as x is not free in $\forall v\mathcal{B}^x_v$ but free for v in $\mathcal{B}^x_v$. First, since $\mathcal{B}^x_v$ has all its free instances of x replaced by v, x is not free in $\forall v\mathcal{B}^x_v$. Second, since $v \neq x$, with the constraint (iii), that v is not free in $\forall x\mathcal{B}$, v is not free in $\mathcal{B}$; so every free instance of v in $\mathcal{B}^x_v$ replaces a free instance of x; so x is free for v in $\mathcal{B}^x_v$. So $\forall v\mathcal{B}^x_v \to \forall x[\mathcal{B}^x_v]^v_x$ is an instance of T3.27. But since v is not free in $\mathcal{B}$, and by constraint (i), v is free for x in $\mathcal{B}$, by T8.2, $[\mathcal{B}^x_v]^v_x = \mathcal{B}$. So $k.2$ is a version of T3.27.

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. This time, there is an application of Gen in Gen* at $k.1$. But A_{k-1} is good and since A_k matches N and, by (ii), v is free in no undischarged auxiliary assumption of N, v is not free in any undischarged auxiliary assumption of A_k; so A_k is good. (Notice that, in this reasoning, we appeal to each of the restrictions that apply to $\forall$I in N).

($\exists$E) If $\mathcal{Q}_k$ arises by $\exists$E, then N looks something like this,

$$\begin{array}{r|l}
h & \exists x\mathcal{B} \\
i & \quad \mathcal{B}^x_v \\
j & \quad \mathcal{C} \\
k & \mathcal{C} \qquad h,i\text{-}j\ \exists\text{E}
\end{array}$$

where $h, i, j < k$, $\exists x\mathcal{B}$ and the subderivation are accessible at line k, and $\mathcal{Q}_k = \mathcal{C}$; further, the *ND* restrictions on ∃E are met: (i) v is free for x in $\mathcal{B}$, (ii) v is not free in any undischarged auxiliary assumption, and (iii) v is not free in $\exists x\mathcal{B}$ or in $\mathcal{C}$. By assumption A_{k-1} matches N through line $k - 1$ and is good. So the formulas at lines h, i and j appear at the same scope on corresponding lines in A_{k-1}; since they appear at the same scope, $\exists x\mathcal{B}$ and the corresponding subderivation are accessible in A_{k-1}. Since A_{k-1} is good, no application of Gen under the scope of $\mathcal{B}^x_v$ is to a variable free in $\mathcal{B}^x_v$. So let A_k continue as follows,

$$\begin{array}{r|ll}
h & \exists x\mathcal{B} & \\
i & \quad \mathcal{B}^x_v & \\
j & \quad \mathcal{C} & \\
k.1 & \mathcal{B}^x_v \to \mathcal{C} & i\text{-}j\ \text{DT} \\
k.2 & \exists v\mathcal{B}^x_v \to \mathcal{C} & k.1\ \text{T3.30} \\
k.3 & \forall v{\sim}\mathcal{B}^x_v \to \forall x{\sim}\mathcal{B} & \text{T3.27} \\
k.4 & (\forall v{\sim}\mathcal{B}^x_v \to \forall x{\sim}\mathcal{B}) \to ({\sim}\forall x{\sim}\mathcal{B} \to {\sim}\forall v{\sim}\mathcal{B}^x_v) & \text{T3.13} \\
k.5 & {\sim}\forall x{\sim}\mathcal{B} \to {\sim}\forall v{\sim}\mathcal{B}^x_v & k.4,k.3\ \text{MP} \\
k.6 & \exists x\mathcal{B} \to \exists v\mathcal{B}^x_v & k.5\ \text{abv} \\
k.7 & \exists v\mathcal{B}^x_v & h,k.6\ \text{MP} \\
k & \mathcal{C} & k.2,k.7\ \text{MP}
\end{array}$$

From constraint (iii), that v is not free in $\mathcal{C}$, $k.2$ meets the restriction on T3.30. If $v = x$ we can go directly from h and $k.2$ to k. So suppose $v \neq x$. Then by [homework] $\forall v{\sim}\mathcal{B}^x_v \to \forall x{\sim}\mathcal{B}$ at $k.3$ is an instance of T3.27. So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. There is an application of Gen in T3.30 at $k.2$. But A_{k-1} is good and since A_k matches N and, by (ii), v is free in no undischarged auxiliary assumption of N, v is not free in any undischarged auxiliary assumption of A_k; so A_k is good. (Notice again that we appeal to each of the restrictions that apply to ∃E in N).

(∃I) Homework.

(=E) Homework.

(=I) Homework.

In any case, A_k matches N through line k and is good.

Indct: Derivation A matches N and is good.

So if there is an *ND* derivation to show $\Gamma \vdash_{ND} \mathcal{P}$, then there is a matching *AD* derivation to show the same; so T9.10, if $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. So with T9.2, *AD* and *ND* are equivalent; that is, $\Gamma \vdash_{ND} \mathcal{P}$ iff $\Gamma \vdash_{AD} \mathcal{P}$. Given this, we will often ignore the difference between *AD* and *ND* and simply write $\Gamma \vdash \mathcal{P}$ when there is a(n *AD* or *ND*) derivation of $\mathcal{P}$ from premises in Γ. Also given the equivalence between the systems, we are in a position to *transfer* results from one system to the other without demonstrating them directly for both. We will come to appreciate this, and especially the relative simplicity of *AD*, as time goes by.

As before, given any *ND* derivation, we can use the method of our induction to find a corresponding *AD* derivation. For a simple example, consider the following demonstration that $\sim A \rightarrow (A \wedge B) \vdash_{ND} A$.

(I)

1.	$\sim A \rightarrow (A \wedge B)$	P
2.	$\quad \sim A$	A (c, $\sim$E)
3.	$\quad A \wedge B$	1,2 $\rightarrow$E
4.	$\quad A$	3 $\wedge$E
5.	$\quad A \wedge \sim A$	4,2 $\wedge$I
6.	A	2-4 $\sim$E

Given relevant cases from the induction, the corresponding *AD* derivation is as follows,

1	$\sim A \to (A \wedge B)$	prem
2	$\quad\mid \sim A$	assp
3	$\quad\mid A \wedge B$	1,2 MP
4.1	$\quad\mid (A \wedge B) \to A$	T3.20
4	$\quad\mid A$	4.1,3 MP
5.1	$\quad\mid A \to (\sim A \to (A \wedge \sim A))$	T9.4
5.2	$\quad\mid \sim A \to (A \wedge \sim A)$	4,5.1 MP
5	$\quad\mid A \wedge \sim A$	5.2,2 MP
6.1	$\sim A \to (A \wedge \sim A)$	2-5 DT
6.2	$(A \wedge \sim A) \to A$	T3.20
6.3	$(A \wedge \sim A) \to \sim A$	T3.19
6.4	$\sim A \to A$	6.1,6.2 T3.2
6.5	$\sim A \to \sim A$	6.1,6.3 T3.2
6.6	$(\sim A \to \sim A) \to ((\sim A \to A) \to A)$	A3
6.7	$(\sim A \to A) \to A$	6.6,6.5 MP
6	A	6.7,6.4 MP

For the first two lines, we simply take over the premise and assumption from the *ND* derivation. For (3), the induction uses MP in *AD* where $\to$E appears in *ND*; so that is what we do. For (4), our induction shows that we can get the effect of $\wedge$E by appeal to T3.20 with MP. (5) in the *ND* derivation is by $\wedge$I, and, as above, we get the same effect by T9.4 with MP. (6) in the *ND* derivation is by $\sim$E. Following the strategy from the induction, we set up for application of A3 by getting the conditional by DT. As usual, the constructed derivation is not very efficient! You should be able to get the same result in just five lines by appeal to T3.20, T3.2 and then T3.7 (try it). But, again, the point is just to show that there always *is* a corresponding derivation.

*E9.12. Set up the above induction for T9.10 and complete the unfinished cases (including the case for $\exists$E) to show that if $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. For cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

E9.13. Consider a system *N2* which is like *ND* except that its only rules are $\wedge$E, $\wedge$I, $\sim$E and $\sim$I, along with the system *A2* from E3.4 on p. 79. Produce a complete demonstration that if $\Gamma \vdash_{N2} \mathcal{P}$, then $\Gamma \vdash_{A2} \mathcal{P}$. You may use any of the theorems for *A2* from E3.4, along with DT from E9.7.

E9.14. Consider the following *ND* derivation and, using the method from the induction, construct a derivation to show $\exists x(C \wedge Bx) \vdash_{AD} C$.

1.	$\exists x(C \wedge Bx)$	P
2.	$\quad C \wedge By$	A (g, 1$\exists$E)
3.	$\quad C$	2 $\wedge$E
4.	C	1,2-3 $\exists$E

Hint: your derivation should have 12 lines.

9.4 Extending to *ND+*

ND+ adds sixteen rules to *ND*: the four inference rules, MT, HS, DS and NB and the twelve replacement rules, DN, Com, Assoc, Idem, Impl, Trans, DeM, Exp, Equiv, Dist, QN and BQN — where some of these have multiple forms. It might seem tedious to go through all the cases but, as it happens, we have already done most of the work. First, it is easy to see that,

T9.11. If $\Gamma \vdash_{ND} \mathcal{P}$ then $\Gamma \vdash_{ND+} \mathcal{P}$.

Suppose $\Gamma \vdash_{ND} \mathcal{P}$. Then there is an *ND* derivation N of $\mathcal{P}$ from premises in Γ. But since every rule of *ND* is a rule of *ND+*, N is a derivation in *ND+* as well. So $\Gamma \vdash_{ND+} \mathcal{P}$.

From T9.2 and T9.11, then, the situation is as follows,

$$\Gamma \vdash_{AD} \mathcal{P} \xrightarrow{9.2} \Gamma \vdash_{ND} \mathcal{P} \xrightarrow{9.11} \Gamma \vdash_{ND+} \mathcal{P}$$

If an argument is valid in *AD*, it is valid in *ND*, and in *ND+*. From T9.10, the leftmost arrow is a biconditional. Again, however, one might think that *ND+* has more resources than *ND*, so that more could be derived in *ND+* than *ND*. But this is not so. To see this, we might begin with the closer systems *ND* and *ND+*, and attempt to show that anything derivable in *ND+* is derivable in *ND*. Alternatively, we choose simply to expand the induction of the previous section to include cases for all the rules of *ND+*. The result is a demonstration that if $\Gamma \vdash_{ND+} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. Given this, the three systems are connected in a "loop" — so that if there is a derivation in any one of the systems, there is a derivation in the others as well.

T9.12. If $\Gamma \vdash_{ND+} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

Suppose $\Gamma \vdash_{ND+} \mathcal{P}$; then there is an *ND+* derivation N of $\mathcal{P}$ from premises in Γ. We show that for any i, there is a good *AD* derivation A_i that matches N through line i.

Basis: The first line of N is a premise or an assumption. Let A_1 be the same. Then A_1 matches N; and since there is no application of Gen, A_1 is good.

Assp: For any i, $0 \leq i < k$, there is a good derivation A_i that matches N through line i.

Show: There is a good derivation of A_k that matches N through line k.

Either $\mathcal{Q}_k$ is a premise or assumption, arises by a rule of *ND*, or by the *ND*+ derivation rules, MT, HS, DS, NB or replacement rules, DN, Com, Assoc, Idem, Impl, Trans, DeM, Exp, Equiv, Dist, QN or BQN. If $\mathcal{Q}_k$ is a premise or assumption or arises by a rule of *ND*, then by reasoning as for T9.10, there is a good derivation A_k that matches N through line k. So suppose $\mathcal{Q}_k$ arises by one of the *ND*+ rules.

(MT) If $\mathcal{Q}_k$ arises from previous lines by MT, then N is something like this,

$$\begin{array}{r|ll} i & \mathcal{B} \rightarrow \mathcal{C} & \\ j & \sim\mathcal{C} & \\ & & \\ k & \sim\mathcal{B} & i,j \text{ MT} \end{array}$$

where $i, j < k$, $\mathcal{B} \rightarrow \mathcal{C}$ and $\sim\mathcal{C}$ are accessible at line k, and $\mathcal{Q}_k = \sim\mathcal{B}$. By assumption A_{k-1} matches N through line $k-1$ and is good. So $\mathcal{B} \rightarrow \mathcal{C}$ and $\sim\mathcal{C}$ appear at the same scope on the lines numbered 'i' and 'j' of A_{k-1} and are accessible in A_{k-1}. So let A_k continue as follows,

$$\begin{array}{r|ll} i & \mathcal{B} \rightarrow \mathcal{C} & \\ j & \sim\mathcal{C} & \\ & & \\ k.1 & (\mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\sim\mathcal{C} \rightarrow \sim\mathcal{B}) & \text{T3.13} \\ k.2 & \sim\mathcal{C} \rightarrow \sim\mathcal{B} & k.1,i \text{ MP} \\ k & \sim\mathcal{B} & k.2,j \text{ MP} \end{array}$$

So $\mathcal{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. And since there is no new application of Gen, A_k is good.

(HS) Homework.

(DS) Homework.

(NB) Homework.

(rep) If If $\mathcal{Q}_k$ arises from a replacement rule *rep* of the form $\mathcal{C} \;\triangleleft\triangleright\; \mathcal{D}$, then N is something like this,

			or			
i	$\mathscr{B}$			i	$\mathscr{B}$	
k	$\mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}$	i rep		k	$\mathscr{B}^{\mathscr{D}}/\!\!/_{\mathscr{C}}$	i rep

where $i < k$, $\mathscr{B}$ is accessible at line k and, in the first case, $\mathscr{Q}_k = \mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}$. By assumption A_{k-1} matches N through line $k-1$ and is good. But by T6.10 - T6.26, T6.29, T6.30, and T6.68, $\vdash_{ND} \mathscr{C} \leftrightarrow \mathscr{D}$; so with T9.10, $\vdash_{AD} \mathscr{C} \leftrightarrow \mathscr{D}$; so by T9.9, $\vdash_{AD} \mathscr{B} \leftrightarrow \mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}$. Call an arbitrary particular result of this sort, *Tx*, and augment A_k as follows,

0.k	$\mathscr{B} \leftrightarrow \mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}$	*Tx*
i	$\mathscr{B}$	
k.1	$(\mathscr{B} \rightarrow \mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}) \wedge (\mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}} \rightarrow \mathscr{B})$	0.k abv
k.2	$[(\mathscr{B} \rightarrow \mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}) \wedge (\mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}} \rightarrow \mathscr{B})] \rightarrow (\mathscr{B} \rightarrow \mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}})$	T3.20
k.3	$\mathscr{B} \rightarrow \mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}$	k.2,k.1 MP
k	$\mathscr{B}^{\mathscr{C}}/\!\!/_{\mathscr{D}}$	k.3,i MP

So $\mathscr{Q}_k$ appears at the same scope on the line numbered 'k' of A_k; so A_k matches N through line k. There may be applications of Gen in the derivation of *Tx*; but that derivation is under the scope of no undischarged assumption. And under the scope of any undischarged assumptions, there is no new application of Gen. So A_k is good. And similarly in the other case, with some work to flip the biconditional $\vdash_{AD} \mathscr{C} \leftrightarrow \mathscr{D}$ to $\vdash_{AD} \mathscr{D} \leftrightarrow \mathscr{C}$.

In any case, A_k matches N through line k and is good.

Indct: Derivation A matches N and is good.

That is it! The key is that work we have already done collapses cases for all the replacement rules into one. So each of the derivation systems, *AD*, *ND*, and *ND+* is equivalent to the others. That is, $\Gamma \vdash_{AD} \mathscr{P}$ iff $\Gamma \vdash_{ND} \mathscr{P}$ iff $\Gamma \vdash_{ND+} \mathscr{P}$. And that is what we set out to show.

*E9.15. Set up the above induction and complete the unfinished cases to show that if $\Gamma \vdash_{ND+} \mathscr{P}$, then $\Gamma \vdash_{AD} \mathscr{P}$. For cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

E9.16. Consider a sentential language with $\sim$ and $\wedge$ primitive, along with systems *N2* with rules $\wedge$E, $\wedge$I, $\sim$E and $\sim$I from E9.13, and *A2* from E3.4 on p. 79.

Theorems of Chapter 9

T9.1 For any ordinary argument $\mathcal{P}_1 \dots \mathcal{P}_n / \mathcal{Q}$, with good translation consisting of II and $\mathcal{P}'_1 \dots \mathcal{P}'_n$, $\mathcal{Q}'$, if $\mathcal{P}'_1 \dots \mathcal{P}'_n \vDash \mathcal{Q}'$, then $\mathcal{P}_1 \dots \mathcal{P}_n / \mathcal{Q}$ is logically valid.

T9.2 If $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vdash_{ND} \mathcal{P}$.

T9.3 (*Deduction Theorem*) If $\Delta \cup \{\mathcal{P}\} \vdash_{AD} \mathcal{Q}$, and no application of Gen under the scope of $\mathcal{P}$ is to a variable free in $\mathcal{P}$, then $\Delta \vdash_{AD} \mathcal{P} \to \mathcal{Q}$.

T9.4 $\vdash_{AD} \mathcal{A} \to (\mathcal{B} \to (\mathcal{A} \wedge \mathcal{B}))$

T9.5 $\vdash_{AD} (\mathcal{A} \to \mathcal{C}) \to [(\mathcal{B} \to \mathcal{C}) \to ((\mathcal{A} \vee \mathcal{B}) \to \mathcal{C})]$

T9.6 For arbitrary terms r, s and t, $\vdash_{AD} (r = s) \to (t = t^{r}/\!\!/_{s})$.

T9.7 For any formula $\mathcal{A}$ and terms r and s, if s is free for the replaced instance of r in $\mathcal{A}$, then $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}^{r}/\!\!/_{s})$.

T9.8 For any formula $\mathcal{A}$ and terms r and s, if s is free for the replaced instances of r in $\mathcal{A}$, then $\vdash_{AD} (r = s) \to (\mathcal{A} \to \mathcal{A}^{r}/_{s})$.

T9.9 For any formulas $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, if $\vdash_{AD} \mathcal{B} \leftrightarrow \mathcal{C}$, then $\vdash_{AD} \mathcal{A} \leftrightarrow \mathcal{A}^{\mathcal{B}}/\!\!/_{\mathcal{C}}$.

T9.10 If $\Gamma \vdash_{ND} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

T9.11 If $\Gamma \vdash_{ND} \mathcal{P}$ then $\Gamma \vdash_{ND+} \mathcal{P}$.

T9.12 If $\Gamma \vdash_{ND+} \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$.

Suppose *N2* is augmented to a system *N2+* that includes rules MT and Com (for $\wedge$). Augment your argument from E9.13 to produce a complete demonstration that if $\Gamma \vdash_{N2+} \mathcal{P}$ then $\Gamma \vdash_{A2} \mathcal{P}$. Hint: You will have to prove some *A2* results parallel to ones for which we have merely appealed to theorems above. Do not forget that you have DT from E9.7.

E9.17. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

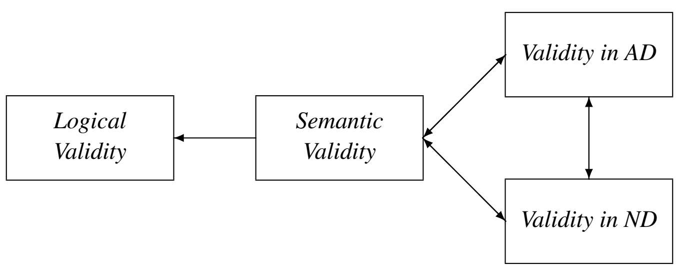
a. The reason semantic validity implies logical validity, but not the other way around.

b. The notion of a *constructive* proof by mathematical induction.

Chapter 10

Main Results

We have introduced four notions of validity, and started to think about their interrelations. In chapter 9, we showed that if an argument is semantically valid, then it is logically valid, and that an argument is valid in *AD* iff it is valid in *ND*. We turn now to the relation between these derivation systems and semantic validity. This completes the project of demonstrating that the different notions of validity are related as follows.



Since *AD* and *ND* are equivalent, it is not necessary separately to establish the relations between *AD* and semantic validity, and between *ND* and semantic validity. Because it is relatively easy to reason about *AD*, we mostly reason about a system like *AD* to establish that an argument is valid in *AD* iff it is semantically valid. From the equivalence between *AD* and *ND* it then follows that an argument is valid in *ND* iff it is semantically valid.

The project divides into two parts. First, we take up the arrows from right to left, and show that if an argument is valid in *AD*, then it is semantically valid: if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. Thus our derivation system is *sound*. If a derivation system is sound, it never leads from premises that are true on an interpretation, to a conclusion

that is not. Second, moving in the other direction, we show that if an argument is semantically valid, then it is valid in *AD*: if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash_{AD} \mathcal{P}$. Thus our derivation system is *adequate*. If a derivation system is adequate, there is a derivation from the premises to the conclusion for every argument that is semantically valid.

10.1 Soundness

It is easy to construct derivation systems that are not sound. Thus, for example, consider a derivation system like *AD* but without the restriction on A4 that the substituted term t be free for the variable x in formula $\mathcal{P}$. Given this, we might reason as follows,

(A)	1.	$\forall x \exists y {\sim}(x = y)$	prem
	2.	$\forall x \exists y {\sim}(x = y) \to \exists y {\sim}(y = y)$	"A4"
	3.	$\exists y {\sim}(y = y)$	1,2 MP

y is not free for x in $\exists y {\sim}(x = y)$; so line (2) is not an instance of A4. And it is a good thing: Consider any interpretation with at least two elements in U. Then it is true that for every x there is some y not identical to it. So the premise is true. But there is no y in U that is not identical to itself. So the conclusion is not true. So the true premise leads to a conclusion that is not true. So the derivation system is not sound.

We would like to show that *AD* is sound — that there is no sequence of moves, no matter how complex or clever, that would lead from premises that are true to a conclusion that is not true. The argument itself is straightforward: suppose $\Gamma \vdash_{AD} \mathcal{P}$; then there is an *AD* derivation $A = \langle \mathcal{Q}_1 \dots \mathcal{Q}_n \rangle$ of $\mathcal{P}$ with $\mathcal{Q}_n = \mathcal{P}$. By induction on line numbers in A, we show that for any i, $\Gamma \vDash \mathcal{Q}_i$. The case when $i = n$ is the desired result. So if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. This general strategy should by now be familiar. However, for the case involving A4, it will be helpful to obtain a pair of preliminary results.

10.1.1 Switching Theorems

In this section, we develop a couple theorems which link substitutions into formulas and terms with substitutions in variable assignments. As we have seen before, the results are a matched pair, with a first result for terms, that feeds into the basis clause for a result about formulas. Perhaps the hardest part is not so much the proofs of the theorems, as understanding what the theorems say. So let us turn to the first.

Suppose we have some terms t and r with interpretation I and variable assignment d. Say $\mathsf{I_d}[r] = \mathsf{o}$. Then the first proposition is this: term t is assigned the same

object on $\mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}$, as $\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}$ is assigned on I_{d}. Intuitively, this is because the same object is fed into the $\mathscr{x}$-place of the term in each case. With $\mathscr{t}$ and $\mathsf{d}(\mathscr{x}|\mathsf{o})$,

$$\text{(B)}\qquad \begin{array}{rl} \mathscr{t}: & \mathscr{h}^n \ldots \mathscr{x} \ldots \\ & \qquad\;\; | \\ \mathsf{d}(\mathscr{x}|\mathsf{o}): & \quad\;\; \ldots \mathsf{o} \ldots \end{array}$$

object o is the input to the "slot" occupied by $\mathscr{x}$. But we are given that $\mathsf{I}_{\mathsf{d}}[\mathscr{r}] = \mathsf{o}$. So with $\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}$ and d,

$$\text{(C)}\qquad \begin{array}{rl} \mathscr{t}^{\mathscr{x}}_{\mathscr{r}}: & \mathscr{h}^n \ldots \mathscr{r} \ldots \\ & \qquad\;\; | \\ \mathsf{d}: & \quad\;\; \ldots \mathsf{o} \ldots \end{array}$$

object o is the input into the "slot" that was occupied by $\mathscr{x}$. So if $\mathsf{I}_{\mathsf{d}}[\mathscr{r}] = \mathsf{o}$, then $\mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}] = \mathsf{I}_{\mathsf{d}}[\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}]$. In the one case, we guarantee that object o goes into the $\mathscr{x}$-place by meddling with the variable assignment. In the other, we get the same result by meddling with the term. Be sure you are clear about this in your own mind. This will be our first result.

T10.1. For any interpretation I, variable assignment d, with terms $\mathscr{t}$ and $\mathscr{r}$, if $\mathsf{I}_{\mathsf{d}}[\mathscr{r}] = \mathsf{o}$, then $\mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}] = \mathsf{I}_{\mathsf{d}}[\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}]$.

For arbitrary terms $\mathscr{t}$ and $\mathscr{r}$, with interpretation I and variable assignment d, suppose $\mathsf{I}_{\mathsf{d}}[\mathscr{r}] = \mathsf{o}$. By induction on the number of function symbols in $\mathscr{t}$, $\mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}] = \mathsf{I}_{\mathsf{d}}[\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}]$.

Basis: If $\mathscr{t}$ has no function symbols, then it is a constant or a variable. Either $\mathscr{t}$ is the variable $\mathscr{x}$ or it is not. (i) Suppose $\mathscr{t}$ is a constant or variable other than $\mathscr{x}$; then $\mathscr{t}^{\mathscr{x}}_{\mathscr{r}} = \mathscr{t}$ (no replacement is made); but d and $\mathsf{d}(\mathscr{x}|\mathsf{o})$ assign just the same things to variables other than $\mathscr{x}$; so they assign just the same things to any variable in $\mathscr{t}$; so by T8.3, $\mathsf{I}_{\mathsf{d}}[\mathscr{t}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}]$. So $\mathsf{I}_{\mathsf{d}}[\mathscr{t}^{x}_{r}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}]$. (ii) If $\mathscr{t}$ is $\mathscr{x}$, then $\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}$ is $\mathscr{r}$ (all of $\mathscr{t}$ is replaced by $\mathscr{r}$); so $\mathsf{I}_{\mathsf{d}}[\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}] = \mathsf{I}_{\mathsf{d}}[\mathscr{r}] = \mathsf{o}$. But $\mathscr{t}$ is $\mathscr{x}$; so $\mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{x}]$; and by TA(v), $\mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{x}] = \mathsf{d}(\mathscr{x}|\mathsf{o})[\mathscr{x}] = \mathsf{o}$. So $\mathsf{I}_{\mathsf{d}}[\mathscr{t}^{x}_{r}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}]$.

Assp: For any i, $0 \le i < k$, for $\mathscr{t}$ with i function symbols, $\mathsf{I}_{\mathsf{d}}[\mathscr{t}^{x}_{r}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}]$.

Show: If $\mathscr{t}$ has k function symbols, then $\mathsf{I}_{\mathsf{d}}[\mathscr{t}^{x}_{r}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}]$.

If $\mathscr{t}$ has k function symbols, then it is of the form, $\mathscr{h}^n \mathscr{s}_1 \ldots \mathscr{s}_n$ where $\mathscr{s}_1 \ldots \mathscr{s}_n$ have $< k$ function symbols. In this case, $\mathscr{t}^{\mathscr{x}}_{\mathscr{r}} = [\mathscr{h}^n \mathscr{s}_1 \ldots \mathscr{s}_n]^{x}_{r}$ $= \mathscr{h}^n \mathscr{s}_1{}^{\mathscr{x}}_{\mathscr{r}} \ldots \mathscr{s}_n{}^{\mathscr{x}}_{\mathscr{r}}$. So $\mathsf{I}_{\mathsf{d}}[\mathscr{t}^{\mathscr{x}}_{\mathscr{r}}] = \mathsf{I}_{\mathsf{d}}[\mathscr{h}^n \mathscr{s}_1{}^{\mathscr{x}}_{\mathscr{r}} \ldots \mathscr{s}_n{}^{\mathscr{x}}_{\mathscr{r}}]$; by TA(f), this is $\mathsf{I}[\mathscr{h}^n]\langle \mathsf{I}_{\mathsf{d}}[\mathscr{s}_1{}^{\mathscr{x}}_{\mathscr{r}}] \ldots \mathsf{I}_{\mathsf{d}}[\mathscr{s}_n{}^{\mathscr{x}}_{\mathscr{r}}]\rangle$. Similarly, $\mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{t}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{h}^n \mathscr{s}_1 \ldots \mathscr{s}_n]$; and by TA(f), this is $\mathsf{I}[\mathscr{h}^n]\langle \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{s}_1] \ldots \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{s}_n]\rangle$. But by assumption, $\mathsf{I}_{\mathsf{d}}[\mathscr{s}_1{}^{\mathscr{x}}_{\mathscr{r}}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{s}_1]$, and …and $\mathsf{I}_{\mathsf{d}}[\mathscr{s}_n{}^{\mathscr{x}}_{\mathscr{r}}] = \mathsf{I}_{\mathsf{d}(\mathscr{x}|\mathsf{o})}[\mathscr{s}_n]$; so

$\langle \mathsf{I_d}[s_1{}^{x}_{r}] \ldots \mathsf{I_d}[s_n{}^{x}_{r}]\rangle = \langle \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_1] \ldots \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_n]\rangle$; so $\mathsf{I}[h^n]\langle \mathsf{I_d}[s_1{}^{x}_{r}]$ $\ldots \mathsf{I_d}[s_n{}^{x}_{r}]\rangle = \mathsf{I}[h^n]\langle \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_1] \ldots \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[s_n]\rangle$; so $\mathsf{I_d}[t^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$.

Indct: For any t, $\mathsf{I_d}[t^x_r] = \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t]$.

Since the "switching" leaves assignments to the parts the same, assignments to the whole remains the same as well.

Similarly, suppose we have we have term r with interpretation I and variable assignment d, where $\mathsf{I_d}[r] = \mathsf{o}$ as before. Suppose r is free for variable x in formula $\mathcal{Q}$. Then the second proposition is that a formula $\mathcal{Q}$ is satisfied on $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}$ iff $\mathcal{Q}^x_r$ is satisfied on $\mathsf{I_d}$. Again, intuitively, this is because the same object is fed into the x-place of the formula in each case. With $\mathcal{Q}$ and $\mathsf{d}(x|\mathsf{o})$,

$$\text{(D)} \qquad \begin{array}{rc} \mathcal{Q}: & \mathcal{Q} \ldots x \ldots \\ & | \\ \mathsf{d}(x|\mathsf{o}): & \ldots \mathsf{o} \ldots \end{array}$$

object o is the input to the "slot" occupied by x. But $\mathsf{I_d}[r] = \mathsf{o}$. So with $\mathcal{Q}^x_r$ and d,

$$\text{(E)} \qquad \begin{array}{rc} \mathcal{Q}^x_r: & \mathcal{Q} \ldots r \ldots \\ & | \\ \mathsf{d}: & \ldots \mathsf{o} \ldots \end{array}$$

object o is the input into the "slot" that was occupied by x. So if $\mathsf{I_d}[r] = \mathsf{o}$ (and r is free for x in $\mathcal{Q}$), then $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$. In the one case, we guarantee that object o goes into the x-place by meddling with the variable assignment. In the other, we get the same result by meddling with the formula. This is our second result, which draws directly upon the first.

T10.2. For any interpretation I, variable assignment d, term r, and formula $\mathcal{Q}$, if $\mathsf{I_d}[r] = \mathsf{o}$, and r is free for x in $\mathcal{Q}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

For arbitrary formula $\mathcal{Q}$, term r and interpretation I, suppose r is free for x in $\mathcal{Q}$. By induction on the number of operator symbols in $\mathcal{Q}$,

Basis: Suppose $\mathsf{I_d}[r] = \mathsf{o}$. If $\mathcal{Q}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \ldots t_n$. In the first case, $\mathcal{Q}^x_r = \mathcal{S}^x_r = \mathcal{S}$. So $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{S}] = \mathsf{S}$; by SF(s), iff $\mathsf{I}[\mathcal{S}] = \mathsf{T}$; by SF(s) again, iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{S}] = \mathsf{S}$; iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$. In the second case, $\mathcal{Q}^x_r = [\mathcal{R}^n t_1 \ldots t_n]^x_r = \mathcal{R}^n t_1{}^x_r \ldots t_n{}^x_r$. So $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{R}^n t_1{}^x_r \ldots t_n{}^x_r] = \mathsf{S}$; by SF(r), iff $\langle \mathsf{I_d}[t_1{}^x_r] \ldots \mathsf{I_d}[t_n{}^x_r]\rangle \in \mathsf{I}[\mathcal{R}^n]$; since $\mathsf{I_d}[r] = \mathsf{o}$, by T10.1, iff $\langle \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t_1] \ldots \mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[t_n]\rangle \in \mathsf{I}[\mathcal{R}^n]$; by SF(r), iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S}$; iff $\mathsf{I}_{\mathsf{d}(x|\mathsf{o})}[\mathcal{Q}] = \mathsf{S}$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{Q}$ has i operator symbols, r is free for x in $\mathcal{Q}$ and $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

Show: If $\mathcal{Q}$ has k operator symbols, r is free for x in $\mathcal{Q}$ and $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

Suppose $\mathsf{I_d}[r] = \mathsf{o}$. If $\mathcal{Q}$ has k operator symbols, then $\mathcal{Q}$ is of the form $\sim\mathcal{B}$, $\mathcal{B} \to \mathcal{C}$, or $\forall v\mathcal{B}$ for variable v and formulas $\mathcal{B}$ and $\mathcal{C}$ with $< k$ operator symbols.

($\sim$) Suppose $\mathcal{Q}$ is $\sim\mathcal{B}$. Then $\mathcal{Q}^x_r = [\sim\mathcal{B}]^x_r = \sim[\mathcal{B}^x_r]$. Since r is free for x in $\mathcal{Q}$, r is free for x in $\mathcal{B}$; so the assumption applies to $\mathcal{B}$. $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_d}[\sim\mathcal{B}^x_r] = \mathsf{S}$; by SF($\sim$), iff $\mathsf{I_d}[\mathcal{B}^x_r] \neq \mathsf{S}$; by assumption iff $\mathsf{I_{d(x|o)}}[\mathcal{B}] \neq \mathsf{S}$; by SF($\sim$), iff $\mathsf{I_{d(x|o)}}[\sim\mathcal{B}] = \mathsf{S}$; iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

($\to$) Homework.

($\forall$) Suppose $\mathcal{Q}$ is $\forall v\mathcal{B}$. Either there are free occurrences of x in $\mathcal{Q}$ or not.

(i) Suppose there are no free occurrences of x in $\mathcal{Q}$. Then $\mathcal{Q}^x_r$ is just $\mathcal{Q}$ (no replacement is made). But since d and $\mathsf{d}(x|\mathsf{o})$ make just the same assignments to variables other than x, they make just the same assignments to all the variables free in $\mathcal{Q}$; so by T8.4, $\mathsf{I_d}[\mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$. So $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

(ii) Suppose there are free occurrences of x in $\mathcal{Q}$. Then x is some variable other than v, and $\mathcal{Q}^x_r = [\forall v\mathcal{B}]^x_r = \forall v[\mathcal{B}^x_r]$.

First, since r is free for x in $\mathcal{Q}$, r is free for x in $\mathcal{B}$, and v is not a variable in r; from this, for any $\mathsf{m} \in \mathsf{U}$, the variable assignments d and $\mathsf{d}(v|\mathsf{m})$ agree on assignments to variables in r; so by T8.3, $\mathsf{I_d}[r] = \mathsf{I_{d(v|m)}}[r]$; so $\mathsf{I_{d(v|m)}}[r] = \mathsf{o}$; so the requirement of the assumption is met for the assignment $\mathsf{d}(v|\mathsf{m})$ and, as an instance of the assumption, for any $\mathsf{m} \in \mathsf{U}$, we have, $\mathsf{I_{d(v|m)}}[\mathcal{B}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(v|m,x|o)}}[\mathcal{B}] = \mathsf{S}$.

Now suppose $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$ but $\mathsf{I_d}[\mathcal{Q}^x_r] \neq \mathsf{S}$; then $\mathsf{I_{d(x|o)}}[\forall v\mathcal{B}] = \mathsf{S}$ but $\mathsf{I_d}[\forall v\mathcal{B}^x_r] \neq \mathsf{S}$. From the latter, by SF($\forall$), there is some $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{I_{d(v|m)}}[\mathcal{B}^x_r] \neq \mathsf{S}$; so by the above result, $\mathsf{I_{d(v|m,x|o)}}[\mathcal{B}] \neq \mathsf{S}$; so by SF($\forall$), $\mathsf{I_{d(x|o)}}[\forall v\mathcal{B}] \neq \mathsf{S}$; this is impossible. And similarly [by homework] in the other direction. So $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$ iff $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$.

If $\mathcal{Q}$ has k operator symbols, if r is free for x in $\mathcal{Q}$ and $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

Indct: For any $\mathcal{Q}$, if r is free for x in $\mathcal{Q}$ and $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

Perhaps the quantifier case looks more difficult than it is. The key point is that since $\mathscr{r}$ is free for x in $\mathcal{Q}$, changes in the assignment to v do not affect the assignment to $\mathscr{r}$. Thus the assumption applies to $\mathcal{B}$ for variable assignments that differ in their assignments to v. This lets us "take the quantifier off," apply the assumption, and then "put the quantifier back on" in the usual way. Another way to make this point is to see how the argument fails when $\mathscr{r}$ is not free for x in $\mathcal{Q}$. If $\mathscr{r}$ is not free for x in $\mathcal{Q}$, then a change in the assignment to v may affect the assignment to $\mathscr{r}$. In this case, although $\mathsf{I_d}[\mathscr{r}] = \mathsf{o}$, $\mathsf{I}_{\mathsf{d}(v|\mathsf{m})}[\mathscr{r}]$ might be something else. So there is no reason to think that substituting $\mathscr{r}$ for x will have the same effect as assigning x to o. As we shall see, this restriction corresponds directly to the one on axiom A4. An example of failure for the axiom is the one (A) with which we began the chapter.

*E10.1. Complete the cases for ($\to$) and ($\forall$) to complete the demonstration of T10.2. You should set up the complete demonstration, but for cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

10.1.2 Soundness

We are now ready for our main proof of soundness for *AD*. Actually, all the parts are already on the table. It is simply a matter of pulling them together into a complete demonstration.

T10.3. If $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. (*Soundness*)

Suppose $\Gamma \vdash_{AD} \mathcal{P}$. Then there is an *AD* derivation $A = \langle \mathcal{Q}_1 \ldots \mathcal{Q}_n \rangle$ of $\mathcal{P}$ from premises in Γ, with $\mathcal{Q}_n = \mathcal{P}$. By induction on the line numbers in A, we show that for any i, $\Gamma \vDash \mathcal{Q}_i$. The case when $i = n$ is the desired result.

Basis: The first line of A is a premise or an axiom. So $\mathcal{Q}_1$ is either a member of Γ or an instance of A1, A2, A3, A4, A5, A6 or A7. The cases for A1, A2, A3, A5, A6 and A7 are parallel.

(prem) If $\mathcal{Q}_1$ is a member of Γ, then there is no interpretation where all the members of Γ are true and $\mathcal{Q}_1$ is not; so by QV, $\Gamma \vDash \mathcal{Q}_1$.

(Ax) Suppose $\mathcal{Q}_1$ is an instance of A1, A2, A3, A5, A6 or A7 and $\Gamma \nvDash \mathcal{Q}_1$. Then by QV, there is some I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{Q}_1] \neq \mathsf{T}$. But by T7.2, T7.3, T7.4, T7.7, T7.8 and T7.9, $\vDash \mathcal{Q}_1$; so by QV, $\mathsf{I}[\mathcal{Q}_1] = \mathsf{T}$. This is impossible, reject the assumption: $\Gamma \vDash \mathcal{Q}_1$.

(A4) If $\mathcal{Q}_1$ is an instance of A4, then it is of the form $\forall x \mathcal{B} \to \mathcal{B}^x_{\mathscr{r}}$ where term $\mathscr{r}$ is free for variable x in formula $\mathcal{B}$. Suppose $\Gamma \nvDash \mathcal{Q}_1$. Then by

QV, there is an I such that $\mathsf{I}[\Gamma] = \mathsf{T}$, but $\mathsf{I}[\forall x \mathcal{B} \to \mathcal{B}^x_r] \neq \mathsf{T}$. From the latter, by TI, there is some d such that $\mathsf{I_d}[\forall x \mathcal{B} \to \mathcal{B}^x_r] \neq \mathsf{S}$; so by SF($\to$), $\mathsf{I_d}[\forall x \mathcal{B}] = \mathsf{S}$ but $\mathsf{I_d}[\mathcal{B}^x_r] \neq \mathsf{S}$; from the first of these, by SF($\forall$), for any $\mathsf{m} \in \mathsf{U}$, $\mathsf{I_{d(x|m)}}[\mathcal{B}] = \mathsf{S}$; in particular, where for some object o, $\mathsf{I_d}[r] = \mathsf{o}$, $\mathsf{I_{d(x|o)}}[\mathcal{B}] = \mathsf{S}$; so, with r free for x in formula $\mathcal{B}$, by T10.2, $\mathsf{I_d}[\mathcal{B}^x_r] = \mathsf{S}$. This is impossible; reject the assumption: $\Gamma \vDash \mathcal{Q}_1$.

Assp: For any i, $1 \leq i < k$, $\Gamma \vDash \mathcal{Q}_i$.

Show: $\Gamma \vDash \mathcal{Q}_k$.

$\mathcal{Q}_k$ is either a premise, an axiom, or arises from previous lines by MP or Gen. If $\mathcal{Q}_k$ is a premise or an axiom then, as in the basis, $\Gamma \vDash \mathcal{Q}_k$. So suppose $\mathcal{Q}_k$ arises by MP or Gen.

(MP) Homework.

(Gen) If $\mathcal{Q}_k$ arises by Gen, then A is something like this,

$i \quad \mathcal{B} \to \mathcal{C}$

$\vdots$

$k \quad \mathcal{B} \to \forall x \mathcal{C} \qquad i$ Gen

where $i < k$, variable x is not free in formula $\mathcal{B}$, and $\mathcal{Q}_k = \mathcal{B} \to \forall x \mathcal{C}$. Suppose $\Gamma \nvDash \mathcal{Q}_k$; then $\Gamma \nvDash \mathcal{B} \to \forall x \mathcal{C}$; so by QV, there is some I such that $\mathsf{I}[\Gamma] = \mathsf{T}$ but $\mathsf{I}[\mathcal{B} \to \forall x \mathcal{C}] \neq \mathsf{T}$; from the latter, by TI, there is a d such that $\mathsf{I_d}[\mathcal{B} \to \forall x \mathcal{C}] \neq \mathsf{S}$; so by SF($\to$), $\mathsf{I_d}[\mathcal{B}] = \mathsf{S}$ but $\mathsf{I_d}[\forall x \mathcal{C}] \neq \mathsf{S}$; from the second of these, by SF($\forall$), there is some $\mathsf{o} \in \mathsf{U}$, such that $\mathsf{I_{d(x|o)}}[\mathcal{C}] \neq \mathsf{S}$. But $\mathsf{I}[\Gamma] = \mathsf{T}$, and by assumption, $\Gamma \vDash \mathcal{B} \to \mathcal{C}$; so by QV, $\mathsf{I}[\mathcal{B} \to \mathcal{C}] = \mathsf{T}$; so by TI, for any variable assignment h, $\mathsf{I_h}[\mathcal{B} \to \mathcal{C}] = \mathsf{S}$; in particular, then, $\mathsf{I_{d(x|o)}}[\mathcal{B} \to \mathcal{C}] = \mathsf{S}$; so by SF($\to$), $\mathsf{I_{d(x|o)}}[\mathcal{B}] \neq \mathsf{S}$ or $\mathsf{I_{d(x|o)}}[\mathcal{C}] = \mathsf{S}$. But since $\mathsf{I_{d(x|o)}}[\mathcal{C}] \neq \mathsf{S}$, we have $\mathsf{I_{d(x|o)}}[\mathcal{B}] \neq \mathsf{S}$; since x is not free in $\mathcal{B}$, d and $\mathsf{d}(x|\mathsf{o})$ agree in their assignments to all the variables free in $\mathcal{B}$; so by T8.4, $\mathsf{I_d}[\mathcal{B}] \neq \mathsf{S}$. This is impossible; reject the assumption: $\Gamma \vDash \mathcal{Q}_k$.

$\Gamma \vDash \mathcal{Q}_k$.

Indct: For any n, $\Gamma \vDash \mathcal{Q}_n$.

So if $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. So *AD* is sound. And since *AD* is sound, with theorems T9.2, T9.11 and T9.12 it follows that *ND* and *ND+* are sound as well.

It is worth commenting on the restriction for Gen. If the restriction fails and x is free in $\mathcal{B}$, then $\mathcal{B} \to \mathcal{C}$ is satisfied on an arbitrary assignment to x just in case each object is such that if it satisfies $\mathcal{B}$ then it satisfies $\mathcal{C}$ — where this may be the case though not every object satisfies $\mathcal{C}$. On the other hand, if x is not free in $\mathcal{B}$, then $\mathcal{B} \to \mathcal{C}$ is satisfied on arbitrary assignments to x just in case if $\mathcal{B}$ is satisfied, then $\mathcal{C}$ is satisfied for each assignment to x. Thus, for example, consider the following derivation which violates the restriction on Gen.

1.	$\forall x(Bx \to Cx)$	prem
2.	$\forall x(Bx \to Cx) \to (Bx \to Cx)$	A4
3.	$Bx \to Cx$	2,1 MP
4.	$Bx \to \forall x Cx$	3 "Gen"

Suppose U is $\mathbb{N}$, the set of all natural numbers, with $\mathsf{I}[\mathcal{B}] = \{\mathsf{o} \mid \mathsf{o} > 5\}$ and $\mathsf{I}[\mathcal{C}] = \{\mathsf{o} | \mathsf{o} > 4\}$. Then every x is such that if it is greater than 5, then it is greater than 4. So the premise is true. But on any assignment that makes x a number greater than 5, it will not be the case that *every* number is greater than 4. So the conclusion is not true. So the derivation system with "Gen" is not sound. This point transfers from Gen to the associated restriction on uses of DT: From the proof of DT, every $\mathcal{Q}_i$ under an auxiliary assumption $\mathcal{B}$ is implicitly of the form $\mathcal{B} \to \mathcal{Q}_i$. So the restriction on Gen naturally transfers to variables free in the assumption $\mathcal{B}$.

*E10.2. Complete the case for (MP) to round out the demonstration that *AD* is sound. You should set up the complete demonstration, but for cases completed in the text, you may simply refer to the text, as the text refers cases to homework.

E10.3. Consider a derivation system *A4* which has axioms and rules,

A4 A1. Any sentential form $\mathcal{P}$ such that $\vDash \mathcal{P}$.

A2. $\vdash \mathcal{P}^x_t \to \exists x \mathcal{P}$ — where t is free for x in $\mathcal{P}$

MP $\mathcal{P} \to \mathcal{Q}, \mathcal{P} \vdash \mathcal{Q}$

$\exists$E $\mathcal{P} \to \mathcal{Q} \vdash \exists x \mathcal{P} \to \mathcal{Q}$ — where x is not free in $\mathcal{Q}$

Provide a complete demonstration that *A4* is sound. You may appeal to substitution results from the text as appropriate. Hint: By the soundness of *AD*, if $\mathcal{P}$ is a sentential form and $\vdash_{AD} \mathcal{P}$ then $\mathcal{P}$ is among axioms of the sort (A1).

10.1.3 Consistency

The proof of soundness is the main result we set out to achieve in this section. But before we go on, it is worth pausing to make an application to *consistency*. Say a set Σ (Sigma) of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \sim\mathcal{A}$. Consistency is thus defined in terms of *derivations* rather than semantic notions. But we show,

T10.4. If there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ (a *model* for Γ), then Γ is consistent.

> Suppose there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ but Γ is inconsistent. From the latter, there is a formula $\mathcal{A}$ such that $\Gamma \vdash \mathcal{A}$ and $\Gamma \vdash \sim\mathcal{A}$; so by T10.3, $\Gamma \vDash \mathcal{A}$ and $\Gamma \vDash \sim\mathcal{A}$. But $\mathsf{M}[\Gamma] = \mathsf{T}$; so by QV, $\mathsf{M}[\mathcal{A}] = \mathsf{T}$ and $\mathsf{M}[\sim\mathcal{A}] = \mathsf{T}$; so by TI, for any d, $\mathsf{M_d}[\mathcal{A}] = \mathsf{S}$ and $\mathsf{M_d}[\sim\mathcal{A}] = \mathsf{S}$; from the second of these, by SF($\sim$), $\mathsf{M_d}[\mathcal{A}] \neq \mathsf{S}$. This is impossible; reject the assumption: if there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$, then Γ is consistent.

This is an interesting and important theorem. Suppose we want to show that some set of formulas is inconsistent. For this, it is enough to *derive* a contradiction from the set. But suppose we want to show that there is no way to derive a contradiction. Merely failing to find a derivation does not show that there is not one! But, with soundness, we can demonstrate that there is no such derivation by finding a model for the set.

Similarly, if we want to show that $\Gamma \vdash \mathcal{A}$, it is enough to *produce* the derivation. But suppose we want to show that $\Gamma \nvdash \mathcal{A}$. Merely failing to find a derivation does not show that there is not one! Still, as above, given soundness, we can demonstrate that there is no derivation by finding a model on which the premises are true, with the negation of the conclusion.

T10.5. If there is an interpretation M such that $\mathsf{M}[\Gamma \cup \{\sim\mathcal{A}\}] = \mathsf{T}$, then $\Gamma \nvdash \mathcal{A}$.

> The reasoning is left for homework. But the idea is very much as above. With soundness, it is impossible to have both $\mathsf{M}[\Gamma \cup \{\sim\mathcal{A}\}] = \mathsf{T}$ and $\Gamma \vdash \mathcal{A}$.

Again, the result is useful. Suppose, for example, we want to show that $\sim\forall x Ax \nvdash \sim Aa$. You may be unable to find a derivation, and be able to point out flaws in a friend's attempt. But we show that there is no derivation by finding a model on which both $\sim\forall x Ax$ and $\sim\sim Aa$ are true. And this is easy. Let $\mathsf{U} = \{1, 2\}$ with $\mathsf{M}[a] = 1$ and $\mathsf{M}[A] = \{1\}$.

(i) Suppose $\mathsf{M}[\sim\forall x Ax] \neq \mathsf{T}$; then by TI, there is some d such that $\mathsf{M_d}[\sim\forall x Ax] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{M_d}[\forall x Ax] = \mathsf{S}$; so by SF($\forall$), for any $\mathsf{o} \in \mathsf{U}$, $\mathsf{M}_{\mathsf{d}(x|\mathsf{o})}[Ax] = \mathsf{S}$; so $\mathsf{M}_{\mathsf{d}(x|2)}[Ax] = \mathsf{S}$. But $\mathsf{d}(x|2)[x] = 2$; so by TA(v), $\mathsf{M}_{\mathsf{d}(x|2)}[x] = 2$; so by SF(r), $2 \in \mathsf{M}[A]$; but $2 \notin \mathsf{M}[A]$. This is impossible; reject the assumption: $\mathsf{M}[\sim\forall x Ax] = \mathsf{T}$. (ii) Suppose $\mathsf{M}[\sim\sim Aa] \neq \mathsf{T}$; then by TI, there is some d such that $\mathsf{M_d}[\sim\sim Aa] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{M_d}[\sim Aa] = \mathsf{S}$; and by SF($\sim$) again, $\mathsf{M_d}[Aa] \neq \mathsf{S}$. But $\mathsf{M}[a] = 1$; so by TA(c), $\mathsf{M_d}[a] = 1$; so by SF(r), $1 \notin \mathsf{M}[A]$; but $1 \in \mathsf{M}[A]$. This is impossible; reject the assumption: $\mathsf{M}[\sim\sim Aa] = \mathsf{T}$. So $\mathsf{M}[\sim\forall x Ax] = \mathsf{T}$ and $\mathsf{M}[\sim\sim Aa] = \mathsf{T}$. So by T10.5, $\sim\forall x Ax \nvDash \sim Aa$.

If there is a model on which all the members of Γ are true and $\sim\mathcal{A}$ is true, then it is not the case that every model with Γ true has $\mathcal{A}$ true. So, with soundness, there cannot be a derivation of $\mathcal{A}$ from Γ.

*E10.4. Provide an argument to show T10.5. Hint: The reasoning is very much as for T10.4.

E10.5. (a) Show that $\{\exists x Ax, \sim Aa\}$ is consistent. (b) Show that $\forall x(Ax \to Bx)$, $\sim Ba \nvDash \sim\exists x Ax$.

10.2 Sentential Adequacy

The proof of soundness is straightforward given methods we have used before. But the proof of adequacy was revolutionary when Gödel first produced it in 1930. It is easy to construct derivation systems that are *not* adequate. Thus, for example, consider a system like the sentential part of *AD* but without A1. It is easy to see that such a system is sound, and so that derivations without A1 do not go astray. (All we have to do is leave the case for A1 out of the proof for soundness.) But, by our discussion of independence from section 11.2 (see also E8.13), there is no derivation of A1 from A2 and A3 alone. So there are sentential expressions $\mathcal{P}$ such that $\vDash \mathcal{P}$, but for which there is no derivation. So the resultant derivation system would not be adequate. We turn now to showing that our derivation systems are in fact adequate: if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. Given this, with soundness, we have $\Gamma \vDash \mathcal{P}$ iff $\Gamma \vdash \mathcal{P}$, so that our derivation systems deliver just the results they are supposed to.

Adequacy for a system like *AD* was first proved by Kurt Gödel in his 1930 doctoral dissertation. The version of the proof that we will consider is the standard one,

essentially due to L. Henkin.[1] An interesting feature of these proofs is that they are not constructive. So far, in proving the equivalence of deductive systems, we have been able to show that there are certain derivations, by showing how to *construct* them. In this case, we show that there are derivations, but without showing how to construct them. As we shall see in Part IV, a constructive proof of adequacy for our full predicate logic is impossible. So this is the only way to go.

The proof of adequacy is more involved than any we have encountered so far. Each of the parts is comparable to what has gone before, and all the parts are straightforward. But there are enough parts that it is possible to lose the forest for the trees. I thus propose to do the proof three times. In this section, we will prove sentential adequacy — that for expressions in a sentential language, if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. This should enable us to grasp the overall shape of the argument without interference from too many details. We will then consider a basic version of the quantificational argument and, after addressing a few complications, put it all together for the full version. Notation and theorem numbers are organized to preserve parallels between the cases.

10.2.1 Basic Idea

The basic idea is straightforward: Let us restrict ourselves to an arbitrary sentential language $\mathcal{L}_s$ and to sentential semantic rules. Derivations are automatically restricted to sentential rules by the restricted language. So derivations and semantics are particularly simple. For formulas in this language, our goal is to show that if $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. We can see how this works with just a couple of preliminaries.

We begin with a definition and a theorem. As before, let us say,

Con A set Σ of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \sim\mathcal{A}$.

So consistency is a syntactical notion. A set of formulas is consistent just in case there is no way to derive a contradiction from it. Now for the theorem,

T10.6$_s$. For any set of formulas Σ and sentence $\mathcal{P}$, if $\Sigma \nvdash \sim\mathcal{P}$, then $\Sigma \cup \{\mathcal{P}\}$ is consistent.

Suppose $\Sigma \nvdash \sim\mathcal{P}$, but $\Sigma \cup \{\mathcal{P}\}$ is not consistent. From the latter, there is some $\mathcal{A}$ such that $\Sigma \cup \{\mathcal{P}\} \vdash \mathcal{A}$ and $\Sigma \cup \{\mathcal{P}\} \vdash \sim\mathcal{A}$. So by DT, $\Sigma \vdash \mathcal{P} \to \mathcal{A}$

[1] Henkin, "Completeness of the First-Order Calculus." Kurt Gödel, "Die Vollständigkeit der Axiome des Logischen Funktionenkalküls." English translation in *From Frege to Gödel*, reprint in *Gödel's Collected Works*.

and $\Sigma \vdash \mathcal{P} \to \sim\mathcal{A}$; by T3.10, $\vdash \sim\sim\mathcal{P} \to \mathcal{P}$; so by T3.2, $\Sigma \vdash \sim\sim\mathcal{P} \to \mathcal{A}$, and $\Sigma \vdash \sim\sim\mathcal{P} \to \sim\mathcal{A}$; but by A3, $\vdash (\sim\sim\mathcal{P} \to \sim\mathcal{A}) \to [(\sim\sim\mathcal{P} \to \mathcal{A}) \to \sim\mathcal{P}]$; so by two instances of MP, $\Sigma \vdash \sim\mathcal{P}$. But this is impossible; reject the assumption: if $\Sigma \nvdash \sim\mathcal{P}$, then $\Sigma \cup \{\mathcal{P}\}$ is consistent.

The idea is simple: if $\Gamma \cup \{\mathcal{P}\}$ is inconsistent, then by reasoning as for $\sim$I in *ND*, $\sim\mathcal{P}$ follows from Γ alone; so if $\sim\mathcal{P}$ cannot be derived from Γ alone, then $\Gamma \cup \{\mathcal{P}\}$ is consistent. Notice that, insofar as the language is sentential, the derivation does not include any applications of Gen, so the applications of DT are sure to meet the restriction on Gen.

In the last section, we saw that any set with a model is consistent. Now suppose we knew the converse, that any consistent set has a model.

($*$) For any consistent set of formulas Σ', there is an interpretation M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

This sets up the key connection between syntactic and semantic notions, between consistency on the one hand, and truth on the other, that we will need for adequacy. Schematically, then, with ($*$) we have the following,

1.	$\Gamma \cup \{\sim\mathcal{P}\}$ has a model	$\Longrightarrow$	$\Gamma \not\models_s \mathcal{P}$	
2.	$\Gamma \cup \{\sim\mathcal{P}\}$ is consistent	$\Longrightarrow$	$\Gamma \cup \{\sim\mathcal{P}\}$ has a model	($*$)
3.	$\Gamma \cup \{\sim\mathcal{P}\}$ is not consistent	$\Longrightarrow$	$\Gamma \vdash \mathcal{P}$	

(2) is just ($*$). (1) is by simple semantic reasoning: Suppose $\Gamma \cup \{\sim\mathcal{P}\}$ has a model; then there is some M such that $\mathsf{M}[\Gamma \cup \{\sim\mathcal{P}\}] = \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\sim\mathcal{P}] = \mathsf{T}$; from the latter, by ST($\sim$), $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so by SV, $\Gamma \not\models_s \mathcal{P}$. (3) is by straightforward syntactic reasoning: Suppose $\Gamma \cup \{\sim\mathcal{P}\}$ is not consistent; then by an application of T10.6_s, $\Gamma \vdash \sim\sim\mathcal{P}$; but by T3.10, $\vdash \sim\sim\mathcal{P} \to \mathcal{P}$; so by MP, $\Gamma \vdash \mathcal{P}$. Now suppose $\Gamma \models_s \mathcal{P}$; then by (1), reading from right to left, $\Gamma \cup \{\sim\mathcal{P}\}$ does not have a model; so by (2), again from right to left, $\Gamma \cup \{\sim\mathcal{P}\}$ is not consistent; so by (3), $\Gamma \vdash \mathcal{P}$. So if $\Gamma \models_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$, which was to be shown. Of course, knowing that there is some way to derive $\mathcal{P}$ is not the same as knowing what that way is. All the same, ($*$) tells us that there must exist a model of a certain sort, from which it follows that there must exist a derivation. And the work of our demonstration of adequacy reduces to a demonstration of ($*$).

So we need to show that every consistent set of formulas Σ' has an interpretation M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$. Here is the basic idea: We show that any consistent Σ' is

a subset of a corresponding "big" set Σ'' specified in such a way that it must have a model M' — which in turn is a model for the smaller Σ'. Following the arrows,

$$\begin{tikzcd}
\Sigma'' \arrow[dr] & \\
& \mathsf{M}' \arrow[dl] \\
\Sigma' \arrow[uu] &
\end{tikzcd}$$

Given a consistent Σ', we show that there is the big set Σ''. From this we show that there must be an M' that is a model not only for Σ'' but for Σ' as well. So if Σ' is consistent, then it has a model. We proceed through a series of theorems to show that this can be done.

10.2.2 Gödel Numbering

In constructing our big sets, we will want to consider formulas, for inclusion or exclusion, serially — one after another. For this, we need to "line them up" for consideration. Thus, in this section we show,

T10.7$_s$. There is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2 \ldots$ of all formulas in $\mathcal{L}_s$.

> The proof is by construction. We develop a method by which the formulas can be lined up. The method is interesting in its own right, and foreshadows methods from Gödel's Incompleteness Theorem for arithmetic.

In subsection 2.1.1, we required that any sentential language $\mathcal{L}_s$ has countably many sentence letters, which can be ordered into a series, $\mathcal{S}_0$, $\mathcal{S}_1 \ldots$. Assume some such series. We want to show that the *formulas* of $\mathcal{L}_s$ can be so ordered as well. Begin by assigning to each symbol α (alpha) in the language an integer $g[\alpha]$, called its *Gödel Number*.

a. $g[(] = 3$

b. $g[)] = 5$

c. $g[\sim] = 7$

d. $g[\rightarrow] = 9$

e. $g[\mathcal{S}_n] = 11 + 2n$

So, for example, $g[\mathscr{S}_0] = 11$ and $g[\mathscr{S}_4] = 11+2\times 4 = 19$. Clearly each symbol gets a unique Gödel number, and Gödel numbers for individual symbols are odd positive integers.

Now we are in a position to assign a Gödel number to each formula as follows: Where $\alpha_0, \alpha_1 \ldots \alpha_n$ are the symbols, in order from left to right, in some expression $\mathcal{Q}$,

$$g[\mathcal{Q}] = 2^{g[\alpha_0]} \times 3^{g[\alpha_1]} \times 5^{g[\alpha_2]} \times \ldots \times \pi_n{}^{g[\alpha_n]}$$

where 2, 3, $5 \ldots \pi_n$ are the first n prime numbers. So, for example, $g[\sim\sim\mathscr{S}_0] = 2^7 \times 3^7 \times 5^{11}$; similarly, $g[\sim(\mathscr{S}_0 \to \mathscr{S}_4)] = 2^7 \times 3^3 \times 5^{11} \times 7^9 \times 11^{19} \times 13^5 = 15463, 36193, 79608, 90364, 71042, 41201, 87066, 87500, 00000$ — a very big integer! All the same, it is an integer, and it is clear that every expression is assigned to some integer.

Further, different expressions get different Gödel numbers. It is a theorem of arithmetic that every integer is uniquely factored into primes (see the arithmetic for Gödel numbering and more arithmetic for Gödel numbering references). So a given integer can correspond to at most one formula: Given a Gödel number, we can find its unique prime factorization; then if there are seven 2s in the factorization, the first symbol is $\sim$; if there are seven 3s, the second symbol is $\sim$; if there are eleven 5s, the third symbol is $\mathscr{S}_0$; and so forth. Notice that numbers for individual *symbols* are odd, where numbers for *expressions* are even (where the number for an atomic comes out odd when it is thought of as a symbol, but then even when it is thought of as a formula).

The point is not that this is a practical, or a fun, procedure. Rather, the point is that we have integers associated with each expression of the language. Given this, we can take the set of all formulas, and *order* its members according to their Gödel numbers — so that there is an enumeration $\mathcal{Q}_1$, $\mathcal{Q}_2 \ldots$ of all formulas. And this is what was to be shown.

E10.6. Find Gödel numbers for the following sentences (for the last, you need not do the calculation).

$\mathscr{S}_7$ $\qquad$ $\sim\mathscr{S}_0$ $\qquad$ $\mathscr{S}_0 \to \sim(\mathscr{S}_1 \to \sim\mathscr{S}_0)$

E10.7. Determine the expressions that have the following Gödel numbers.

49 $\qquad$ 1944 $\qquad$ $2^7 \times 3^3 \times 5^{11} \times 7^9 \times 11^7 \times 13^{13} \times 17^5$

Some Arithmetic Relevant to Gödel Numbering

Say an integer i has a "representation as a product of primes" if there are some primes $p_a, p_b \ldots p_j$ such that $p_a \times p_b \times \ldots \times p_j = i$. We understand a single prime p to be its own representation.

G1. Every integer > 1 has at least one representation as a product of primes.

Basis: 2 is prime and so is its own representation; so the first integer > 1 has a representation as a product of primes.

Assp: For any i, $1 < i < k$, i has a representation as a product of primes.

Show: k has a representation as a product of primes.

If k is prime, the result is immediate; so suppose there are some $i, j < k$ such that $k = i \times j$; by assumption i has a representation as a product of primes $p_a \times \ldots \times p_b$ and j has a representation as a product of primes $q_a \times \ldots \times q_b$; so $k = i \times j = p_a \times \ldots \times p_b \times q_a \times \ldots \times q_b$ has a representation as a product of primes.

Indct: Any $i > 1$ has a representation as a product of primes.

Corollary: any integer > 1 is divided by at least one prime.

G2. There are infinitely many prime numbers.

Suppose the number of primes is finite; then there is some list $p_1, p_2 \ldots p_n$ of all the primes; consider $q = p_1 \times p_2 \times \ldots \times p_n + 1$; no p_i in the list $p_1 \ldots p_n$ divides q evenly, since each leaves remainder 1; but by the corollary to (G1), q is divided by some prime; so some prime is not on the list; reject the assumption: there are infinitely many primes.

Note: Sometimes q, calculated this way, is itself prime: when the list is $\{2\}$, $q = 2 + 1 = 3$, and 3 is prime. Similarly, $2 \times 3 + 1 = 7$, $2 \times 3 \times 5 + 1 = 31$, $2 \times 3 \times 5 \times 7 + 1 = 211$, and $2 \times 3 \times 5 \times 7 \times 11 + 1 = 2311$, where 7, 31, 211, and 2311 are all prime. But $2 \times 3 \times 5 \times 7 \times 11 \times 13 + 1 = 30031 = 59 \times 509$. So we are not always *finding* a prime not on the list, but rather only showing that there *is* a prime not on it.

G3. For any $i > 1$, if i is the product of the primes $p_1, p_2 \ldots p_a$, then no distinct collection of primes $q_1, q_2 \ldots q_b$ is such that i is the product of them. (The *Fundamental Theorem* of Arithmetic)

For a proof, see the more arithmetic for Gödel numbering reference in the corresponding part of the next section.

E10.8. Which would come first in the official enumeration of formulas, $\mathcal{S}_1 \to \sim\mathcal{S}_2$ or $\mathcal{S}_2 \to \sim\mathcal{S}_2$? Explain. Hint: you should be able to do this without actually calculating the Gödel numbers.

10.2.3 The Big Set

Recall that a set Σ is consistent iff there is no $\mathcal{A}$ such that Σ implies both $\mathcal{A}$ and $\sim\mathcal{A}$. Now, a set Σ is *maximal* iff for any $\mathcal{A}$ the set implies one or the other.

Max A set Σ of formulas is *maximal* iff for any sentence $\mathcal{A}$, $\Sigma \vdash \mathcal{A}$ or $\Sigma \vdash \sim\mathcal{A}$.

Again, this is a syntactical notion. If a set is maximal, then it implies $\mathcal{A}$ or $\sim\mathcal{A}$ for any sentence $\mathcal{A}$; if it is consistent, then it does not imply both. We set out to construct a big set Σ'' from Σ', and show that Σ'' is both maximal and consistent.

CnsΣ'' Construct Σ'' from Σ' as follows: By T10.7_s, there is an enumeration, $\mathcal{Q}_1$, $\mathcal{Q}_2$... of all the formulas in $\mathcal{L}_s$. Consider this enumeration, and let Ω_0 (Omega$_0$) be the same as Σ'. Then for any $i > 0$, let

$$\Omega_i = \Omega_{i-1} \qquad \text{if} \qquad \Omega_{i-1} \vdash \sim\mathcal{Q}_i$$

else,

$$\Omega_i = \Omega_{i-1} \cup \{\mathcal{Q}_i\} \qquad \text{if} \qquad \Omega_{i-1} \nvdash \sim\mathcal{Q}_i$$

then,

$$\Sigma'' = \bigcup_{i \geq 0} \Omega_i \text{ — that is, } \Sigma'' \text{ is the union of all the } \Omega_i\text{s}$$

Beginning with set Σ' $(= \Omega_0)$, we consider the formulas in the enumeration $\mathcal{Q}_1$, $\mathcal{Q}_2$... one-by-one, adding a formula to the set just in case its negation is not already derivable. Σ'' contains all the members of Σ' together with all the formulas added this way. Observe that $\Sigma' \subseteq \Sigma''$. One might think of the Ω_is as constituting a big "sack" of formulas, and the $\mathcal{Q}_i$s as coming along on a conveyor belt: for a given $\mathcal{Q}_i$, if there is no way to derive its negation from formulas already in the sack, we throw the $\mathcal{Q}_i$ in; otherwise, we let it go on by. Of course, this is not a procedure we could complete in finite time. Rather, we give a *logical* condition which specifies, for any $\mathcal{Q}_i$ in the language, whether it is to be included in Σ'' or not. The important point is that some Σ'' meeting these conditions *exists*.

As an example, suppose $\Sigma' = \{\sim A \to B\}$ and consider an enumeration which begins A, $\sim A$, B, $\sim B$.... Then,

$\Omega_0 = \Sigma'$; so $\Omega_0 = \{\sim A \to B\}$.

$\mathcal{Q}_1 = A$, and $\Omega_0 \nvdash \sim A$; so $\Omega_1 = \{\sim A \to B\} \cup \{A\} = \{\sim A \to B, A\}$.

(F) $\mathcal{Q}_2 = \sim A$, and $\Omega_1 \vdash \sim\sim A$; and Ω_2 is unchanged; so $\Omega_2 = \{\sim A \to B, A\}$.

$\mathcal{Q}_3 = B$, and $\Omega_2 \nvdash \sim B$; so $\Omega_3 = \{\sim A \to B, A\} \cup \{B\} = \{\sim A \to B, A, B\}$.

$\mathcal{Q}_4 = \sim B$, and $\Omega_3 \vdash \sim\sim B$; and Ω_4 is unchanged; so $\Omega_4 = \{\sim A \to B, A, B\}$.

So we include $\mathcal{Q}_i$ each time its negation is not implied. Ultimately, we will use this set to construct a model. For now, though, the point is simply to understand the condition under which a formula is included or excluded from the set.

We now show that if Σ' is consistent, then Σ'' is maximal and consistent. Perhaps the first is obvious: We guarantee that Σ'' is maximal by including $\mathcal{Q}_i$ as a member whenever $\sim\mathcal{Q}_i$ is not already a consequence.

T10.8$_\mathrm{s}$. If Σ' is consistent, then Σ'' is maximal and consistent.

The proof comes to the demonstration of three results. Given the assumption that Σ' is consistent, we show, (a) Σ'' is maximal; (b) each Ω_i is consistent; and use this to show (c), Σ'' is consistent. Suppose Σ' is consistent.

(a) Σ'' is maximal. Suppose otherwise. Then there is some $\mathcal{Q}_i$ such that both $\Sigma'' \nvdash \mathcal{Q}_i$ and $\Sigma'' \nvdash \sim\mathcal{Q}_i$. For this i, by construction, each member of Ω_{i-1} is in Σ''; so if $\Omega_{i-1} \vdash \sim\mathcal{Q}_i$ then $\Sigma'' \vdash \sim\mathcal{Q}_i$; but $\Sigma'' \nvdash \sim\mathcal{Q}_i$; so $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$; so by construction, $\Omega_i = \Omega_{i-1} \cup \{\mathcal{Q}_i\}$; and by construction again, $\mathcal{Q}_i \in \Sigma''$; so $\Sigma'' \vdash \mathcal{Q}_i$. This is impossible; reject the assumption: Σ'' is maximal.

(b) Each Ω_i is consistent. By induction on the series of Ω_is.

Basis: $\Omega_0 = \Sigma'$ and Σ' is consistent; so Ω_0 is consistent.

Assp: For any i, $0 \leq i < k$, Ω_i is consistent.

Show: Ω_k is consistent.

Ω_k is either Ω_{k-1} or $\Omega_{k-1} \cup \{\mathcal{Q}_k\}$. Suppose the former; by assumption, Ω_{k-1} is consistent; so Ω_k is consistent. Suppose the latter; then by construction, $\Omega_{k-1} \nvdash \sim\mathcal{Q}_k$; so by T10.6$_s$, $\Omega_{k-1} \cup \{\mathcal{Q}_k\}$ is consistent; so Ω_k is consistent. So, either way, Ω_k is consistent.

Indct: For any i, Ω_i is consistent.

(c) Σ'' is consistent. Suppose Σ'' is not consistent; then there is some $\mathcal{A}$ such that $\Sigma'' \vdash \mathcal{A}$ and $\Sigma'' \vdash \sim\mathcal{A}$. Consider derivations *D1* and *D2* of these results,

and the premises $\mathcal{Q}_i \dots \mathcal{Q}_j$ of these derivations. Where $\mathcal{Q}_j$ is the last of these premises in the enumeration of formulas, by the construction of Σ'', each of $\mathcal{Q}_i \dots \mathcal{Q}_j$ must be a member of Ω_j; so *D1* and *D2* are derivations from Ω_j; so Ω_j is inconsistent. But by the previous result, Ω_j is consistent. This is impossible; reject the assumption: Σ'' is consistent.

Because derivations of $\mathcal{A}$ and $\sim\mathcal{A}$ have only finitely many premises, all the premises in a derivation of a contradiction must show up in some Ω_j; so if Σ'' is inconsistent, then some Ω_j is inconsistent. But no Ω_j is inconsistent. So Σ'' is consistent. So we have what we set out to show. $\Sigma' \subseteq \Sigma''$, and if Σ' is consistent, then Σ'' is both maximal and consistent.

E10.9. (i) Suppose $\Sigma' = \{A \to \sim B\}$ and the enumeration of formulas begins A, $\sim A$, B, $\sim B \dots$ What are Ω_0, Ω_1, Ω_2, Ω_3, and Ω_4? (ii) What are they when the enumeration begins B, $\sim B$, A, $\sim A \dots$? In each case, produce a (sentential) model to show that the resultant Ω_4 is consistent.

10.2.4 The Model

We now construct a model M' for Σ'. In this sentential case, the specification is particularly simple.

CnsM′ For any atomic $\mathcal{S}$, let $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$.

Notice that there clearly exists some such interpretation M': We assign T to every sentence letter that can be derived from Σ'', and F to the others. It will not be the case that we are in a position to do all the derivations, and so to know what are all the assignments to the atomics. Still, it must be that any atomic either is or is not a consequence of Σ', and so that there exists a corresponding interpretation M' on which those sentence letters either are or are not assigned T.

We now want to show that if Σ' is consistent, then M' is a model for Σ' — that if Σ' is consistent then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. As we shall see, this results immediately from the following theorem.

T10.9$_s$. If Σ' is consistent, then for any sentence $\mathcal{B}$, of $\mathcal{L}_s$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Suppose Σ' is consistent. Then by T10.8$_s$, Σ'' is maximal and consistent. Now by induction on the number of operators in $\mathcal{B}$,

Basis: If $\mathcal{B}$ has no operators, then it is an atomic of the sort $\mathcal{S}$. But by the construction of M', $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$; so $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{B}$ has i operator symbols, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Show: If $\mathcal{B}$ has k operator symbols, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

If $\mathcal{B}$ has k operator symbols, then it is of the form $\sim\mathcal{P}$ or $\mathcal{P} \rightarrow \mathcal{Q}$ where $\mathcal{P}$ and $\mathcal{Q}$ have $< k$ operator symbols.

($\sim$) Suppose $\mathcal{B}$ is $\sim\mathcal{P}$. (i) Suppose $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$; then $\mathsf{M}'[\sim\mathcal{P}] = \mathsf{T}$; so by ST($\sim$), $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so by assumption, $\Sigma'' \nvdash \mathcal{P}$; so by maximality, $\Sigma'' \vdash \sim\mathcal{P}$; which is to say, $\Sigma'' \vdash \mathcal{B}$. (ii) Suppose $\Sigma'' \vdash \mathcal{B}$; then $\Sigma'' \vdash \sim\mathcal{P}$; so by consistency, $\Sigma'' \nvdash \mathcal{P}$; so by assumption, $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{M}'[\sim\mathcal{P}] = \mathsf{T}$; which is to say, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$. So $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

($\rightarrow$) Suppose $\mathcal{B}$ is $\mathcal{P} \rightarrow \mathcal{Q}$. (i) Suppose $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$; then $\mathsf{M}'[\mathcal{P} \rightarrow \mathcal{Q}] = \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$ or $\mathsf{M}'[\mathcal{Q}] = \mathsf{T}$; so by assumption, $\Sigma'' \nvdash \mathcal{P}$ or $\Sigma'' \vdash \mathcal{Q}$. Suppose the latter; by A1, $\vdash \mathcal{Q} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})$; so by MP, $\Sigma'' \vdash \mathcal{P} \rightarrow \mathcal{Q}$. Suppose the former; then by maximality, $\Sigma'' \vdash \sim\mathcal{P}$; but by T3.9, $\vdash \sim\mathcal{P} \rightarrow (\mathcal{P} \rightarrow \mathcal{Q})$; so by MP, $\Sigma'' \vdash \mathcal{P} \rightarrow \mathcal{Q}$. So in either case, $\Sigma'' \vdash \mathcal{P} \rightarrow \mathcal{Q}$; where this is to say, $\Sigma'' \vdash \mathcal{B}$. (ii) Suppose $\Sigma'' \vdash \mathcal{B}$ but $\mathsf{M}'[\mathcal{B}] \neq \mathsf{T}$; by [homework], this is impossible: so if $\Sigma'' \vdash \mathcal{B}$, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$. So $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

If $\mathcal{B}$ has k operator symbols, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Indct: For any $\mathcal{B}$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

So if Σ' is consistent, then for any $\mathcal{B} \in \Sigma''$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

The key to this is that Σ'' is both maximal and consistent. In (F), for example, $\Omega_0 = \{\sim A \rightarrow B\}$; so $\Omega_0 \nvdash A$ and $\Omega_0 \nvdash B$; if we were simply to follow our construction procedure as applied to this set, the result would have $\mathsf{M}'[A] \neq \mathsf{T}$ and $\mathsf{M}'[B] \neq \mathsf{T}$; but then $\mathsf{M}'[\sim A \rightarrow B] \neq \mathsf{T}$ and there is no model for Ω_0. But Ω_4 has A and B as members; so $\Omega_4 \vdash A$ and $\Omega_4 \vdash B$. So by the construction procedure, $\mathsf{M}'[A] = \mathsf{T}$ and $\mathsf{M}'[B] = \mathsf{T}$; so $\mathsf{M}'[\sim A \rightarrow B] = \mathsf{T}$. Thus it is the construction with maximality and consistency of Σ'' that puts us in a position to draw the parallel between the implications of Σ'' and what is true on M'. It is now a short step to seeing that we have a model for Σ' and so ($*$) that we have been after.

*E10.10. Complete the second half of the conditional case to complete the proof of T10.9_s. You should set up the entire induction, but may refer to the text for parts completed there, as the text refers to homework.

E10.11. (i) Where $\Sigma' = \{A \to \sim B\}$, and the enumeration of formulas are as in the first part of E10.9, what assignments does M' make to A and B? (ii) What assignments does it make on the second enumeration? Use a truth table to show, for each case, that the assignments result in a *model* for Σ'. Explain.

10.2.5 Final Result

The proof of sentential adequacy is now a simple matter of pulling together what we have done. First, it is a simple matter to show,

T10.10$_s$. If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. $(*)$

Suppose Σ' is consistent but $\mathsf{M}'[\Sigma'] \neq \mathsf{T}$. From the latter, there is some formula $\mathcal{B} \in \Sigma'$ such that $\mathsf{M}'[\mathcal{B}] \neq \mathsf{T}$. Since $\mathcal{B} \in \Sigma'$, by construction, $\mathcal{B} \in \Sigma''$; so $\Sigma'' \vdash \mathcal{B}$; so, since Σ' is consistent, by T10.9$_s$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$. This is impossible; reject the assumption: if Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

That is it! Going back to the beginning of our discussion of sentential adequacy, all we needed was $(*)$, and now we have it. So the final argument is as sketched before:

T10.11$_s$. If $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. (*sentential adequacy*)

Suppose $\Gamma \vDash_s \mathcal{P}$ but $\Gamma \nvdash \mathcal{P}$. Say, for the moment, that $\Gamma \vdash \sim\sim\mathcal{P}$; by T3.10, $\vdash \sim\sim\mathcal{P} \to \mathcal{P}$; so by MP, $\Gamma \vdash \mathcal{P}$; but this is impossible; so $\Gamma \nvdash \sim\sim\mathcal{P}$. Given this, by T10.6$_s$, $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent; so by T10.10$_s$, there is a model M' such that $\mathsf{M}'[\Gamma \cup \{\sim\mathcal{P}\}] = \mathsf{T}$; so $\mathsf{M}'[\sim\mathcal{P}] = \mathsf{T}$; so by ST($\sim$), $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}'[\Gamma] = \mathsf{T}$ but $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so by SV, $\Gamma \nvDash_s \mathcal{P}$. This is impossible; reject the assumption: if $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$.

Try again to get the complete picture in your mind: The key is that consistent sets always have models. If there is no derivation of $\mathcal{P}$ from Γ, then $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent; and if $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent, then it has a model — so that $\Gamma \nvDash_s \mathcal{P}$. Thus, put the other way around, if $\Gamma \vDash_s \mathcal{P}$, then there is a derivation of $\mathcal{P}$ from Γ. We get the key point, that consistent sets have models, by finding a relation between consistent, and *maximal* consistent sets. If a set is both maximal and consistent, then it contains enough information about its atomics that a model for its atomics is a model for the whole.

It is obvious that the argument is not constructive — we do not see how to show that $\Gamma \vdash \mathcal{P}$ whenever $\Gamma \vDash_s \mathcal{P}$. But it is interesting to see why. The argument turns on the *existence* of our big sets under certain conditions, and so on the existence of

models. We show that the sets must exist and have certain properties, though we are not in a position to find all their members. This puts us in a position to know the existence of derivations, though we do not say what they are.[2]

E10.12. Suppose our primitive operators are $\sim$ and $\wedge$ and the derivation system is *A2* from E3.4 on p. 79. Present a complete demonstration of adequacy for this derivation system — with all the definitions and theorems. You may simply appeal to the text for results that require no change.

10.3 Quantificational Adequacy: Basic Version

As promised, the demonstration of quantificational adequacy is parallel to what we have seen. Return to a quantificational language and to our regular quantificational semantic and derivation notions. The goal is to show that if $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. Certain complications are avoided if we suppose that the language $\mathcal{L}'$ includes infinitely many constants not in Γ, and does not include the '$=$' symbol for equality. The constants not already in Γ are required for the construction of our big sets. And without $=$ in the language, the model specification is simplified. We will work through the basic argument in this section and, dropping constraints on the language, return to the general case in the next. If you are confused at any stage, it may help to refer back to the parallel section for the sentential case.

Before launching into the main argument, it will be helpful to have a preliminary theorem. Where $D = \langle \mathcal{B}_1 \ldots \mathcal{B}_n \rangle$ is an *AD* derivation, and $\Sigma' = \{\mathcal{C}_1 \ldots \mathcal{C}_n\}$ is a set of formulas, for some constant a and variable x, say $D^{a}_{x} = \langle \mathcal{B}_1{}^{a}_{x} \ldots \mathcal{B}_n{}^{a}_{x} \rangle$ and $\Sigma'{}^{a}_{x} = \{\mathcal{C}_1{}^{a}_{x} \ldots \mathcal{C}_n{}^{a}_{x}\}$. By induction on the line numbers in D, we show,

T10.12. If D is a derivation from Σ', and x is a variable that does not appear in D, then for any constant a, D^{a}_{x} is a derivation from $\Sigma'{}^{a}_{x}$.

Basis: $\mathcal{B}_1$ is either a member of Σ' or an axiom.

(prem) If $\mathcal{B}_1$ is a member of Σ', then $\mathcal{B}_1{}^{a}_{x}$ is a member of $\Sigma'{}^{a}_{x}$; so $\langle \mathcal{B}_1{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

(eq) If $\mathcal{B}_1$ is an equality axiom, A5, A6 or A7, then it includes no constants; so $\mathcal{B}_1 = \mathcal{B}_1{}^{a}_{x}$; so $\mathcal{B}_1{}^{a}_{x}$ is an equality axiom, and $\langle \mathcal{B}_1{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

[2] In fact, there are constructive approaches to sentential adequacy. See, for example, Lemma 1.13 and Proposition 1.14 of Mendelson, *Introduction to Mathematical Logic*. Our primary purpose, however, is to set up the argument for the quantificational case, where such methods do not apply.

(A1) If $\mathcal{B}_1$ is an instance of A1, then it is of the form, $\mathcal{P} \to (\mathcal{Q} \to \mathcal{P})$; so $\mathcal{B}_1{}^{a}_{x}$ is $\mathcal{P}^a_x \to (\mathcal{Q}^a_x \to \mathcal{P}^a_x)$; but this is an instance of A1; so if $\mathcal{B}_1$ is an instance of A1, then $\mathcal{B}_1{}^{a}_{x}$ is an instance of A1, and $\langle \mathcal{B}_1{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

(A2) Homework.

(A3) Homework.

(A4) If $\mathcal{B}_1$ is an instance of A4, then it is of the form, $\forall v\mathcal{P} \to \mathcal{P}^v_t$, for some variable v and term t that is free for v in $\mathcal{P}$. So $\mathcal{B}_1{}^{a}_{x} = [\forall v\mathcal{P} \to \mathcal{P}^v_t]^a_x = [\forall v\mathcal{P}]^a_x \to [\mathcal{P}^v_t]^a_x$. But since x does not appear in D, $x \neq v$; so $[\forall v\mathcal{P}]^a_x = \forall v[\mathcal{P}^a_x]$. And by T8.7, $[\mathcal{P}^v_t]^a_x = [\mathcal{P}^a_x]^v_{t^a_x}$. So $\mathcal{B}_1{}^{a}_{x} = \forall v[\mathcal{P}^a_x] \to [\mathcal{P}^a_x]^v_{t^a_x}$; and since x is new to D and t is free for v in $\mathcal{P}$, t^a_x is free for v in $\mathcal{P}^a_x$; so $\forall v[\mathcal{P}^a_x] \to [\mathcal{P}^a_x]^v_{t^a_x}$ is an instance of A4; so if $\mathcal{B}_1$ is an instance of A4, then $\mathcal{B}_1{}^{a}_{x}$ is an instance of A4, and $\langle \mathcal{B}_1{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

Assp: For any i, $1 \leq i < k$, $\langle \mathcal{B}_1{}^{a}_{x} \dots \mathcal{B}_i{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

Show: $\langle \mathcal{B}_1{}^{a}_{x} \dots \mathcal{B}_k{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

$\mathcal{B}_k$ is a member of Σ', an axiom, or arises from previous lines by MP or Gen. If $\mathcal{B}_k$ is a member of Σ' or an axiom then, by reasoning as in the basis, $\langle \mathcal{B}_1 \dots \mathcal{B}_k \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$. So two cases remain.

(MP) Homework.

(Gen) If $\mathcal{B}_k$ arises by Gen, then there are some lines in D,

$i \quad \mathcal{P} \to \mathcal{Q}$

$\quad\vdots$

$k \quad \mathcal{P} \to \forall v\mathcal{Q} \qquad i$ Gen

where $i < k$, v is not free in $\mathcal{P}$ and $\mathcal{B}_k = \mathcal{P} \to \forall v\mathcal{Q}$. By assumption $(\mathcal{P} \to \mathcal{Q})^a_x$ is a member of the derivation $\langle \mathcal{B}_1{}^{a}_{x} \dots \mathcal{B}_{k-1}{}^{a}_{x} \rangle$ from $\Sigma'{}^{a}_{x}$; but $(\mathcal{P} \to \mathcal{Q})^a_x$ is $\mathcal{P}^a_x \to \mathcal{Q}^a_x$; and since x does not appear in D, it cannot be that x is the same variable as v; so v is not free in $\mathcal{P}^a_x$; so $\mathcal{P}^a_x \to \forall v\mathcal{Q}^a_x = [\mathcal{P} \to \forall v\mathcal{Q}]^a_x$ follows in this new derivation by Gen. So $\langle \mathcal{B}_1{}^{a}_{x} \dots \mathcal{B}_k{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

So $\langle \mathcal{B}_1{}^{a}_{x} \dots \mathcal{B}_k{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

Indct: For any n, $\langle \mathcal{B}_1{}^{a}_{x} \dots \mathcal{B}_n{}^{a}_{x} \rangle$ is a derivation from $\Sigma'{}^{a}_{x}$.

The reason this works is that none of the justifications change: switching x for a leaves each line justified for the same reasons as before. The only sticking point

may be the case for A4. But we did the real work for this by induction in T8.7. And that result should be intuitive, once we see what it says. Given this, the rest is straightforward.

*E10.13. Finish the cases for A2, A3 and MP to complete the proof of T10.12. You should set up the complete demonstration, but may refer to the text for cases completed there, as the text refers cases to homework.

E10.14. Where $\Sigma' = \{Ab\}$ and D is as follows,

1.	$\forall x{\sim}Ax \rightarrow {\sim}Ab$	A4
2.	$(\forall x{\sim}Ax \rightarrow {\sim}Ab) \rightarrow ({\sim}{\sim}Ab \rightarrow {\sim}\forall x{\sim}Ax)$	T3.13
3.	${\sim}{\sim}Ab \rightarrow {\sim}\forall x{\sim}Ax$	2,1 MP
4.	$Ab \rightarrow {\sim}{\sim}Ab$	T3.11
5.	$Ab \rightarrow {\sim}\forall x{\sim}Ax$	4,3 T3.2
6.	Ab	prem
7.	${\sim}\forall x{\sim}Ax$	5,6 MP
8.	$\exists x Ax$	7 abv

apply T10.12 to show that D^b_y is a derivation from $\Sigma'^{\,b}_{\,y}$. Do any of the justifications change? Explain.

10.3.1 Basic Idea

As before, our main argument turns on the idea that every consistent set has a model. Thus we begin with a definition and a theorem.

Con A set Σ of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash {\sim}\mathcal{A}$.

So a set of formulas is consistent just in case there is no way to derive a contradiction from it. Of course, now we are working with full quantificational languages, and so with our complete quantificational derivation systems.

For the following theorem, notice that Σ is a set of *formulas*, and $\mathcal{P}$ a *sentence* (a distinction without a difference in the sentential case). Again as before,

T10.6. For any set of formulas Σ and sentence $\mathcal{P}$, if $\Sigma \nvdash {\sim}\mathcal{P}$, then $\Sigma \cup \{\mathcal{P}\}$ is consistent.

For some sentence $\mathcal{P}$, suppose $\Sigma \nvdash {\sim}\mathcal{P}$ but $\Sigma \cup \{\mathcal{P}\}$ is not consistent. From the latter, there is some formula $\mathcal{A}$ such that $\Sigma \cup \{\mathcal{P}\} \vdash \mathcal{A}$ and $\Sigma \cup \{\mathcal{P}\} \vdash$

> $\sim\mathscr{A}$; since $\mathscr{P}$ is a sentence, it has no free variables; so by DT, $\Sigma \vdash \mathscr{P} \to \mathscr{A}$ and $\Sigma \vdash \mathscr{P} \to \sim\mathscr{A}$; by T3.10, $\vdash \sim\sim\mathscr{P} \to \mathscr{P}$; so by T3.2, $\Sigma \vdash \sim\sim\mathscr{P} \to \mathscr{A}$ and $\Sigma \vdash \sim\sim\mathscr{P} \to \sim\mathscr{A}$; but by A3, $\vdash (\sim\sim\mathscr{P} \to \sim A) \to [(\sim\sim\mathscr{P} \to \mathscr{A}) \to \sim\mathscr{P}]$; so by two instances of MP, $\Sigma \vdash \sim\mathscr{P}$. This is impossible; reject the assumption: if $\Sigma \nvdash \sim\mathscr{P}$, then $\Sigma \cup \{\mathscr{P}\}$ is consistent.

Insofar as $\mathscr{P}$ is required to be a sentence, the restriction on applications of DT is sure to be met: since $\mathscr{P}$ has no free variables, no application of Gen is to a variable free in $\mathscr{P}$. So T10.6 does not apply to arbitrary formulas.

To the extent that T10.6 plays a direct role in our basic argument for adequacy, this point that it does not apply to arbitrary formulas might seem to present a problem about reaching our general result, that if $\Gamma \vDash \mathscr{P}$ then $\Gamma \vdash \mathscr{P}$, which is supposed to apply in the arbitrary case. But there is a way around the problem. For any formula $\mathscr{P}$, let its (*universal*) *closure* $\mathscr{P}^c$ be $\mathscr{P}$ prefixed by a universal quantifier for every variable free in $\mathscr{P}$. To make $\mathscr{P}^c$ unique, for some enumeration of variables, $x_1, x_2 \ldots$ let the quantifiers be in order of ascending subscripts. So if $\mathscr{P}$ has no free variables, $\mathscr{P}^c = \mathscr{P}$; if x_1 is free in $\mathscr{P}$, then $\mathscr{P}^c = \forall x_1\mathscr{P}$; if x_1 and x_3 are free in $\mathscr{P}$, then $\mathscr{P}^c = \forall x_1\forall x_3\mathscr{P}$; and so forth. So for any formula $\mathscr{P}$, $\mathscr{P}^c$ is a *sentence*. As it turns out, we will be able to argue about arbitrary formulas $\mathscr{P}$, by using their closures $\mathscr{P}^c$ as intermediaries.

Suppose that the members of $\Gamma \cup \{\sim\mathscr{P}^c\} = \Sigma'$ are formulas of $\mathscr{L}'$. Then it will be sufficient for us to show that any consistent set of this sort has a model.

($\star$) For any consistent set Σ' of formulas in $\mathscr{L}'$, there is an interpretation M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

Again, this sets up the key connection between syntactic and semantic notions — between consistency on the one hand, and truth on the other — that we will need for adequacy. Supposing ($\star$) we have the following,

1.	$\Gamma \cup \{\sim\mathscr{P}^c\}$ has a model	$\Longrightarrow$	$\Gamma \nvDash \mathscr{P}$	
2.	$\Gamma \cup \{\sim\mathscr{P}^c\}$ is consistent	$\Longrightarrow$	$\Gamma \cup \{\sim\mathscr{P}^c\}$ has a model	($\star$)
3.	$\Gamma \cup \{\sim\mathscr{P}^c\}$ is not consistent	$\Longrightarrow$	$\Gamma \vdash \mathscr{P}$	

(2) is just ($\star$). Observe that (1) and (3) switch between $\mathscr{P}^c$ and $\mathscr{P}$. (1) is by semantic reasoning: Suppose $\Gamma \cup \{\sim\mathscr{P}^c\}$ has a model; then there is some M such that $\mathsf{M}[\Gamma \cup \{\sim\mathscr{P}^c\}] = \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\sim\mathscr{P}^c] = \mathsf{T}$; from the latter, by TI, for arbitrary d, $\mathsf{M_d}[\sim\mathscr{P}^c] = \mathsf{S}$; so by SF($\sim$), $\mathsf{M_d}[\mathscr{P}^c] \neq \mathsf{S}$; so by TI, $\mathsf{M}[\mathscr{P}^c] \neq \mathsf{T}$; so by repeated

applications of T7.6 on page 367, $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so by QV, $\Gamma \not\models \mathcal{P}$. (3) is by syntactic reasoning: Suppose $\Gamma \cup \{\sim\mathcal{P}^c\}$ is not consistent; then since $\mathcal{P}^c$ is a sentence, by an application of T10.6, $\Gamma \vdash \sim\sim\mathcal{P}^c$; but by T3.10, $\vdash \sim\sim\mathcal{P}^c \rightarrow \mathcal{P}^c$; so by MP, $\Gamma \vdash \mathcal{P}^c$; and by repeated applications of A4 and MP, $\Gamma \vdash \mathcal{P}$.

Now suppose $\Gamma \models \mathcal{P}$; then from (1), $\Gamma \cup \{\sim\mathcal{P}^c\}$ does not have a model; so by (2), $\Gamma \cup \{\sim\mathcal{P}^c\}$ is not consistent; so by (3), $\Gamma \vdash \mathcal{P}$. So if $\Gamma \models \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$, and this is the result we want. T7.6, according to which $\mathsf{M}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{M}[\forall x\mathcal{P}] = \mathsf{T}$, along with A4 and Gen*, which let us derive $\mathcal{P}$ from $\forall x\mathcal{P}$ and vice versa, bridge between $\mathcal{P}$ and $\mathcal{P}^c$ so that our suppositions about formulas can be converted into claims about sentences and then back again.

Again, it remains to show $(\star)$, that every consistent set Σ' of formulas has a model. And, again, our strategy is to find a "big" set related to Σ' which can be used to specify a model for Σ'.

10.3.2 Gödel Numbering

As before, in constructing our big sets, we will want to line up expressions serially — one after another. The method merely expands our approach for the sentential case.

T10.7. There is an enumeration $\mathcal{Q}_1$, $\mathcal{Q}_2 \ldots$ of all the formulas, terms, and the like, in $\mathcal{L}'$.

> The proof is again by construction: We develop a method by which all the expressions of $\mathcal{L}'$ can be lined up. Then the collection of all formulas, taken in that order, is an an enumeration of all formulas; the collection of all terms, taken in that order, is an enumeration of all terms; and so forth.

Insofar as the collections of variable symbols, constant symbols, function symbols, sentence letters, and relation symbols in any quantificational language are countable, they are capable of being sorted into series, x_0, $x_1 \ldots$ and a_0, $a_1 \ldots$ and h^n_0, $h^n_1 \ldots$ and $\mathcal{R}^n_0$, $\mathcal{R}^n_1 \ldots$ for variables, constants, function symbols and relation symbols, respectively (where we think of sentence letters as 0-place relation symbols). Supposing that they are sorted into such series, begin by assigning to each symbol α in $\mathcal{L}'$ an integer $g[\alpha]$ called its *Gödel Number*.

a. $g[(] = 3$
b. $g[)] = 5$
c. $g[\sim] = 7$
d. $g[\rightarrow] = 9$
f. $g[\forall] = 13$
g. $g[x_i] = 15 + 10i$
h. $g[a_i] = 17 + 10i$
i. $g[h^n_i] = 19 + 10(2^n \times 3^i)$

*e. $g[=] = 11$ j. $g[\mathcal{R}^n_i] = 21 + 10(2^n \times 3^i)$

Officially, we do not yet have '$=$' in the language, but it is easy enough to leave it out for now. So, for example, $g[x_0] = 15$, $g[x_1] = 15 + 10 \times 1 = 25$, and $g[\mathcal{R}^2_1] = 21 + 10(2^2 \times 3^1) = 141$.

To see that each symbol gets a distinct Gödel number, first notice th at numbers in different categories cannot overlap: Each of (a) - (f) is obviously distinct and ≤ 13. But (g) - (j) are all greater than 13, and when divided by 10, the remainder is 5 for variables, 7 for constants 9 for function symbols, and 1 for relation symbols; so variables, constants, and function symbols all get different numbers. Second, different symbols get different numbers within the categories. This is obvious except in cases (i) and (j). For these we need to see that each n/i combination results in a different multiplier.

> Suppose this is not so, that there are some combinations n, i and m, j such that $2^n \times 3^i = 2^m \times 3^j$ but $n \neq m$ or $i \neq j$. If $n = m$ then, dividing both sides by 2^n, we get $3^i = 3^j$, so that $i = j$. So suppose $n \neq m$ and, without loss of generality, that $n > m$. Dividing each side by 2^m and 3^i, we get $2^{n-m} = 3^{j-i}$; since $n > m$, $n - m$ is a positive integer; so 2^{n-m} is > 1 and even. But 3^{i-j} is either < 1 or odd. Reject the assumption: if $2^n \times 3^i = 2^m \times 3^j$, then $n = m$ and $i = j$.

So each n/i combination gets a different multiplier, and we conclude that each symbol gets a different Gödel number. (This result is a special case of the Fundamental theorem of Arithmetic treated in the arithmetic fore Gödel numbering and more arithmetic for Gödel numbering references.)

Now, as before, assign Gödel numbers to expressions as follows: Where $\alpha_0, \alpha_1 \ldots \alpha_n$ are the symbols, in order from left to right, in some expression $\mathcal{Q}$,

$$g[\mathcal{Q}] = 2^{g[\alpha_0]} \times 3^{g[\alpha_1]} \times 5^{g[\alpha_2]} \times \ldots \times \pi_n{}^{g[\alpha_n]}$$

where 2, 3, 5... π_n are the first n prime numbers. So, for example, $g[\sim\sim\mathcal{R}^2_1 x_0 x_1] = 2^7 \times 3^7 \times 5^{141} \times 7^{15} \times 11^{25}$ — a relatively large integer (one with over 130 digits)! All the same, it is an integer, and different expressions get different Gödel numbers. Given a Gödel number, we can find the corresponding expression by finding its prime factorization; then if there are seven 2s in the factorization, the first symbol is $\sim$; if there are seven 3s, the second symbol is $\sim$; if there are one hundred forty one 5s, the third symbol is $\mathcal{R}^2_1$; and so forth. Notice that numbers for individual symbols are odd, where numbers for expressions are even.

So we can take the set of all formulas, the set of all terms, or whatever, and order their members according to their Gödel numbers — so that there is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2 \ldots$ of all formulas, terms, and so forth. And this is what was to be shown.

More Arithmetic Relevant to Gödel Numbering

G3. For any $i > 1$, if i is the product of the primes $p_1, p_2 \ldots p_a$, then no distinct collection of primes $q_1, q_2 \ldots q_b$ is such that i is the product of them. (The *Fundamental Theorem* of Arithmetic)

Basis: The first integer $\geq 1 = 2$; but the only collection of primes such that their product is equal to 2 is the collection containing just 2 itself; so no distinct collection of primes is such that 2 is the product of them.

Assp: For any i, $1 \leq i < k$, if i is the product of primes $p_1 \ldots p_a$, then no distinct collection of primes $q_1 \ldots q_b$ is such that i is the product of them.

Show: k is such that if it is the product of the primes $p_1 \ldots p_a$, then no distinct collection of primes $q_1 \ldots q_b$ is such that k is the product of them.

Suppose there are distinct collections of primes $p_1 \ldots p_a$ and $q_1 \ldots q_b$ such that $k = p_1 \times \ldots \times p_a = q_1 \times \ldots \times q_b$; divide out terms common to both lists of primes; then for some subclasses of the original lists, $n = p_1 \times \ldots \times p_c = q_1 \times \ldots \times q_d$, where no member of $p_1 \ldots p_c$ is a member of $q_1 \ldots q_d$ and *vice versa* (of course this p_1 may be distinct from the one in the original list, and so forth). So $p_1 \neq q_1$; suppose, without loss of generality, that $p_1 > q_1$; and let $m = q_1(n/q_1 - n/p_1) = n - (q_1/p_1)n = n - q_1 \times p_2 \times \ldots \times p_c$.

Some preliminary results: (i) $m < n \leq k$; so $m < k$. Further, n/q_1 and n/p_1 are integers, with the first greater than the second; so the difference is an integer > 0; any prime is > 1; so q_1 is > 1; so the product of q_1 and $(n/q_1 - n/p_1)$ is > 1; so $m > 1$. So the inductive assumption applies to m. (ii) q_1 divides n and q_1 divides $q_1 \times p_2 \times \ldots \times p_c$; so $[n - q_1 \times p_2 \times \ldots \times p_c]/q_1$ is an integer; so m/q_1 is an integer, and q_1 divides m. (iii) $(p_1 - q_1)/q_1 = p_1/q_1 - 1$; since p_1 is prime, this is no integer; so q_1 does not divide $(p_1 - q_1)$.

Notice that $m = (p_1 - q_1)(n/p_1)$; either $p_1 - q_1 = 1$ or it has some prime factorization, and n/p_1 has a prime factorization, $p_2 \times \ldots \times p_c$; the product of the factorization(s) is a prime factorization of m. Given the cancellation of common terms to get n, q_1 is not a member of $p_2 \times \ldots \times p_c$; by (iii), q_1 is not a member of the factorization of $p_1 - q_1$; so q_1 is not a member of this factorization of m. By (ii), q_1 divides m, and however many times it goes into m, by (G1), that number has a prime factorization; the product of q_1 and this factorization is a prime factorization of m; so q_1 is a member of some prime factorization of m. But by (i), the inductive assumption applies to m; so m has only one prime factorization. Reject the assumption: there are no distinct collections of primes, $p_1 \ldots p_a$ and $q_1 \ldots q_b$ such that $k = p_1 \times \ldots \times p_a = q_1 \times \ldots \times q_b$.

Indct: For any $i > 1$, if i is the product of the primes $p_1, p_2 \ldots p_a$, then no distinct collection of primes $q_1, q_2 \ldots q_b$ is such that i is the product of them.

E10.15. Find Gödel numbers for each of the following. Treat the first as a simple symbol. (For the last, you need not do the calculation!)

$$\mathcal{R}^2_3 \qquad h^1_1 x_1 \qquad \forall x_2 \mathcal{R}^2_1 a_2 x_2$$

E10.16. Determine the objects that have the following Gödel numbers.

$$61 \qquad 2^{13} \times 3^{15} \times 5^3 \times 7^{15} \times 11^{11} \times 13^{15} \times 17^5$$

10.3.3 The Big Set

This section, along with the next, constitutes the heart of our demonstration of adequacy. Last time, to build our big set we added formulas to Σ' to form a Σ'' that was both maximal and consistent. A set of formulas is consistent just in case there is no formula $\mathcal{A}$ such that both $\mathcal{A}$ and $\sim\mathcal{A}$ are consequences. To accommodate restrictions from T10.6, maximality is defined in terms of *sentences*.

Max A set Σ of formulas is *maximal* iff for any sentence $\mathcal{A}$, $\Sigma \vdash \mathcal{A}$ or $\Sigma \vdash \sim\mathcal{A}$.

This time, however, we need an additional property for our big sets. If a maximal and consistent set has $\forall x\mathcal{P}$ as a member, then it has $\mathcal{P}^x_a$ as a consequence for every constant a. (Be clear about why this is so.) But in a maximal and consistent set, the status of a universal $\forall x\mathcal{P}$ is not always reflected at the level of its instances. Thus, for example, though a set has $\mathcal{P}^x_a$ as a consequence for every constant a, it may consistently include $\sim\forall x\mathcal{P}$ as well — for it may be that a universal is falsified by some individual to which no constant is assigned. But when we come to showing by induction that there is a model for our big set, it will be important that the status of a universal *is* reflected at the level of its instances. We guarantee this by building the set to satisfy the following condition.

Scgt A set Σ of formulas is a *scapegoat* set iff for any sentence $\sim\forall x\mathcal{P}$, if $\Sigma \vdash \sim\forall x\mathcal{P}$, then there is some constant a such that $\Sigma \vdash \sim\mathcal{P}^x_a$.

Equivalently, Σ is a scapegoat set just in case any sentence $\exists x\mathcal{P}$ is such that if $\Sigma \vdash \exists x\mathcal{P}$, then there is some constant a such that $\Sigma \vdash \mathcal{P}^x_a$. In a scapegoat set, we assert the existence of a particular individual (a *scapegoat*) corresponding to any existential claim. Notice that, since $\sim\forall x\mathcal{P}$ is a sentence, $\sim\mathcal{P}^x_a$ is a sentence too.

So we set out to construct from Σ' a maximal, consistent, scapegoat set. As before, the idea is to line the formulas up, and consider them for inclusion one-by-one. In addition, this time, we consider an enumeration of constants c_1, $c_2 \ldots$ and

for any included sentence of the form $\sim\forall x\mathcal{P}$, we include $\sim\mathcal{P}^x_c$ where c is a constant that does not so far appear in the construction. Notice that if, as we have assumed, $\mathcal{L}'$ includes infinitely many constants not in Γ, there are sure to *be* infinitely many constants not already in a Σ' built on Γ.

CnsΣ'' Construct Σ'' from Σ' as follows: By T10.7, there is an enumeration, $\mathcal{Q}_1$, $\mathcal{Q}_2$... of all the sentences in $\mathcal{L}'$ and also an enumeration $c_1, c_2 \ldots$ of constants not in Σ'. Let $\Omega_0 = \Sigma'$. Then for any $i > 0$, let

$\Omega_i = \Omega_{i-1}$	if	$\Omega_{i-1} \vdash \sim\mathcal{Q}_i$
else,		
$\Omega_{i*} = \Omega_{i-1} \cup \{\mathcal{Q}_i\}$	if	$\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$
and,		
$\Omega_i = \Omega_{i*}$	if	$\mathcal{Q}_i$ is not of the form $\sim\forall x\mathcal{P}$
$\Omega_i = \Omega_{i*} \cup \{\sim\mathcal{P}^x_c\}$	if	$\mathcal{Q}_i$ is of the form $\sim\forall x\mathcal{P}$; c the first constant not in Ω_{i*}

then,

$\Sigma'' = \bigcup_{i\geq 0}\Omega_i$ — that is, Σ'' is the union of all the Ω_is

Beginning with set Σ' ($= \Omega_0$), we consider the sentences in the enumeration $\mathcal{Q}_1$, $\mathcal{Q}_2 \ldots$ one-by-one, adding a sentence just in case its negation is not already derivable. In addition, if $\mathcal{Q}_i$ is of the sort $\sim\forall x\mathcal{P}$, we add an instance of it, using a new constant. This time, Ω_{i*} functions as an intermediate set. Observe that if c is not in Ω_{i*}, then c is not in $\sim\forall x\mathcal{P}$. Σ'' contains all the members of Σ', together with all the formulas added this way.

It remains to show that if Σ' is consistent, then Σ'' is a maximal, consistent, scapegoat set.

T10.8. If Σ' is consistent, then Σ'' is a maximal, consistent, scapegoat set.

The proof comes to showing (a) Σ'' is maximal. (b) If Σ' is consistent then each Ω_i is consistent. From this, (c) if Σ' is consistent then Σ'' is consistent. And (d) if Σ' is consistent, then Σ'' is a scapegoat set. Suppose Σ' is consistent.

(a) Σ'' is maximal. Suppose Σ'' is not maximal. Then there is some sentence $\mathcal{Q}_i$ such that both $\Sigma'' \nvdash \mathcal{Q}_i$ and $\Sigma'' \nvdash \sim\mathcal{Q}_i$. For this i, by construction, each member of Ω_{i-1} is in Σ''; so if $\Omega_{i-1} \vdash \sim\mathcal{Q}_i$ then $\Sigma'' \vdash \sim\mathcal{Q}_i$; but $\Sigma'' \nvdash \sim\mathcal{Q}_i$; so $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$; so by construction, $\Omega_{i*} = \Omega_{i-1} \cup \{\mathcal{Q}_i\}$; and

by construction again, $\mathcal{Q}_i \in \Sigma''$; so $\Sigma'' \vdash \mathcal{Q}_i$. This is impossible; reject the assumption: Σ'' is maximal.

(b) Each Ω_i is consistent. By induction on the series of Ω_is.

Basis: $\Omega_0 = \Sigma'$ and Σ' is consistent; so Ω_0 is consistent.

Assp: For any i, $0 \leq i < k$, Ω_i is consistent.

Show: Ω_k is consistent.

Ω_k is either (i) Ω_{k-1}, (ii) $\Omega_{k^*} = \Omega_{k-1} \cup \{\mathcal{Q}_k\}$, or (iii) $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\}$.

(i) Suppose Ω_k is Ω_{k-1}. By assumption, Ω_{k-1} is consistent; so Ω_k is consistent.

(ii) Suppose Ω_k is $\Omega_{k^*} = \Omega_{k-1} \cup \{\mathcal{Q}_k\}$. Then by construction, $\Omega_{k-1} \nvdash \sim\mathcal{Q}_k$; so, since $\mathcal{Q}_k$ is a sentence, by T10.6, $\Omega_{k-1} \cup \{\mathcal{Q}_k\}$ is consistent; so Ω_{k^*} is consistent, and Ω_k is consistent.

(iii) Suppose Ω_k is $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\}$ for c not in Ω_{k^*} or in $\sim\forall x\mathcal{P}$. In this case, as in (ii) above, Ω_{k^*} is consistent; and, by construction $\sim\forall x\mathcal{P} \in \Omega_{k^*}$; so $\Omega_{k^*} \vdash \sim\forall x\mathcal{P}$. Suppose Ω_k is inconsistent; then there are formulas $\mathcal{A}$ and $\sim\mathcal{A}$ such that $\Omega_k \vdash \mathcal{A}$ and $\Omega_k \vdash \sim\mathcal{A}$; so $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\} \vdash \mathcal{A}$ and $\Omega_{k^*} \cup \{\sim\mathcal{P}^x_c\} \vdash \sim\mathcal{A}$. But since $\sim\mathcal{P}^x_c$ is a sentence, the restriction on DT is met, and both $\Omega_{k^*} \vdash \sim\mathcal{P}^x_c \rightarrow \mathcal{A}$ and $\Omega_{k^*} \vdash \sim\mathcal{P}^x_c \rightarrow \sim\mathcal{A}$; by A3, $\vdash (\sim\mathcal{P}^x_c \rightarrow \sim\mathcal{A}) \rightarrow [(\sim\mathcal{P}^x_c \rightarrow \mathcal{A}) \rightarrow \mathcal{P}^x_c]$; so by two instances of MP, $\Omega_{k^*} \vdash \mathcal{P}^x_c$.

Consider some derivation of this result; by T10.12, we can switch c for some variable v that does not occur in Ω_{k^*} or in the derivation, and the result is a derivation; so $\Omega_{k^*}{}^{c}_{v} \vdash [\mathcal{P}^x_c]^c_v$; but since c does not occur in Ω_{k^*} or in $\sim\forall x\mathcal{P}$, this is to say, $\Omega_{k^*} \vdash \mathcal{P}^x_v$; so by Gen*, $\Omega_{k^*} \vdash \forall v\mathcal{P}^x_v$; but x is not free in $\forall v\mathcal{P}^x_v$ and x is free for v in $\mathcal{P}^x_v$, so by T3.27, $\vdash \forall v\mathcal{P}^x_v \rightarrow \forall x[\mathcal{P}^x_v]^v_x$; so by MP, $\Omega_{k^*} \vdash \forall x[\mathcal{P}^x_v]^v_x$; and since v is not a variable in $\mathcal{P}$, it is not free in $\mathcal{P}$ and free for x in $\mathcal{P}$; so by T8.2, $[\mathcal{P}^x_v]^v_x = \mathcal{P}$; so $\Omega_{k^*} \vdash \forall x\mathcal{P}$.

But $\Omega_{k^*} \vdash \sim\forall x\mathcal{P}$. So Ω_{k^*} is inconsistent. This is impossible; reject the assumption: Ω_k is consistent.

Ω_k is consistent

Indct: For any i, Ω_i is consistent.

(c) Σ'' is consistent. Suppose Σ'' is not consistent; then there is some $\mathcal{A}$ such that $\Sigma'' \vdash \mathcal{A}$ and $\Sigma'' \vdash \sim\mathcal{A}$. Consider derivations *D1* and *D2* of these results,

and the premises $\mathcal{Q}_i \ldots \mathcal{Q}_j$ of these derivations. Where $\mathcal{Q}_j$ is the last of these premises in the enumeration of formulas, by the construction of Σ'', each of $\mathcal{Q}_i \ldots \mathcal{Q}_j$ must be a member of Ω_j; so *D1* and *D2* are derivations from Ω_j; so Ω_j is inconsistent. But by the previous result, Ω_j is consistent. This is impossible; reject the assumption: Σ'' is consistent.

(d) Σ'' is a scapegoat set. Suppose $\Sigma'' \vdash \mathcal{Q}_i$, for $\mathcal{Q}_i$ of the form $\sim\forall x\mathcal{P}$. By (c), Σ'' is consistent; so $\Sigma'' \nvdash \sim\sim\forall x\mathcal{P}$; which is to say, $\Sigma'' \nvdash \sim\mathcal{Q}_i$; so, $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$; so by construction, $\Omega_{i*} = \Omega_{i-1} \cup \{\sim\forall x\mathcal{P}\}$ and $\Omega_i = \Omega_{i*} \cup \{\sim\mathcal{P}^x_c\}$; so by construction, $\sim\mathcal{P}^x_c \in \Sigma''$; so $\Sigma'' \vdash \sim\mathcal{P}^x_c$. So if $\Sigma'' \vdash \sim\forall x\mathcal{P}$, then $\Sigma'' \vdash \sim\mathcal{P}^x_c$, and Σ'' is a scapegoat set.

In a pattern that should be familiar by now, we guarantee maximal scapegoat sets, by including instances as required. The most difficult case is (iii) for consistency. Having shown that $\Omega_{k*} \vdash \mathcal{P}^x_c$ for c not in Ω_{k*} or in $\mathcal{P}$, we want to generalize to show that $\Omega_{k*} \vdash \forall x\mathcal{P}$. But, in our derivation systems, generalization is on variables, not constants. To get the generalization we want, we first use T10.12 to replace c with an arbitrary variable v. From this, we might have moved immediately to $\forall x\mathcal{P}$ by the *ND* rule $\forall$I. However, in the above reasoning, we stick with the pattern of *AD* rules, applying Gen*, and then T3.27 to switch bound variables, for the desired result, that contradicts $\sim\forall x\mathcal{P}$.

E10.17. Let $\Sigma' = \{\forall x{\sim}Bx, Ca\}$ and consider enumerations of sentences and extra constants in $\mathcal{L}'$ that begin, Aa, Ba, $\sim\forall xCx \ldots$ and $c_1, c_2 \ldots$. What are Ω_0, Ω_{1*}, Ω_1, Ω_{2*}, Ω_2, Ω_{3*}, Ω_3? Produce a model to show that the resultant set Ω_3 is consistent.

E10.18. Suppose some $\Omega_{i-1} = \{Ac_2, \forall x(Ax \to Bx)\}$. Show that Ω_{i*} is consistent, but Ω_i is not, if $\mathcal{Q}_i = \sim\forall xBx$, and we add $\sim\forall xBx$ with $\sim Bc_2$ to form Ω_{i*} and Ω_i. Why cannot this happen in the construction of Σ''?

10.3.4 The Model

We turn now to constructing the model M' for Σ'. As it turns out, the construction is simplified by our assumption that '=' does not appear in the language. A quantificational interpretation has a universe, with assignments to sentence letters, constants, function symbols, and relation symbols.

CnsM$'$ Let the universe U be the set of positive integers, $\{1, 2\ldots\}$. Then, where a *variable-free* term consists just of function symbols and constants, consider an enumeration t_1, $t_2\ldots$ of all the variable-free terms in $\mathscr{L}'$. If t_z is a constant, set $\mathsf{M}'[t_z] = \mathsf{z}$. If $t_z = h^n t_a \ldots t_b$ for some function symbol h^n and n variable-free terms $t_a \ldots t_b$, then let $\langle\langle \mathsf{a} \ldots \mathsf{b}\rangle, \mathsf{z}\rangle \in \mathsf{M}'[h^n]$. For a sentence letter $\mathscr{S}$, let $\mathsf{M}'[\mathscr{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathscr{S}$. And for a relation symbol $\mathscr{R}^n$, let $\langle \mathsf{a} \ldots \mathsf{b}\rangle \in \mathsf{M}'[\mathscr{R}^n]$ iff $\Sigma'' \vdash \mathscr{R}^n t_a \ldots t_b$.[3]

Thus, for example, where t_1 and t_3 from the enumeration of terms are constants and $\Sigma'' \vdash \mathscr{R}t_1t_3$, then $\mathsf{M}'[t_1] = 1$, $\mathsf{M}'[t_3] = 3$ and $\langle 1, 3\rangle \in \mathsf{M}'[\mathscr{R}]$. Given this, it should be clear *why* $\mathscr{R}t_1t_3$ comes out satisfied on M': Put generally, where $t_a \ldots t_b$ are constants, we set $\mathsf{M}'[t_a] = \mathsf{a}$, and ... and $\mathsf{M}'[t_b] = \mathsf{b}$; so by TA(c), for any variable assignment d, $\mathsf{M}'_\mathsf{d}[t_a] = \mathsf{a}$, and ... and $\mathsf{M}'_\mathsf{d}[t_b] = \mathsf{b}$. So by SF(r), $\mathsf{M}'_\mathsf{d}[\mathscr{R}^n t_a \ldots t_b] = \mathsf{S}$ iff $\langle \mathsf{a} \ldots \mathsf{b}\rangle \in \mathsf{M}'[\mathscr{R}^n]$; by construction, iff $\Sigma'' \vdash \mathscr{R}^n t_a \ldots t_b$. Just as in the sentential case, our idea is to make atomic sentences true on M' just in case they are proved by Σ''.

Our aim has been to show that if Σ' is consistent, then Σ' has a model. We have constructed an interpretation M', and now show what sentences are true on it. As in the sentential case, the main weight is carried by a preliminary theorem. And, as in the sentential case, the key is that we can appeal to special features of Σ'', this time that it is a maximal, consistent, scapegoat set. Notice that $\mathscr{B}$ is a *sentence*.

T10.9. If Σ' is consistent, then for any sentence $\mathscr{B}$ of $\mathscr{L}'$, $\mathsf{M}'[\mathscr{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathscr{B}$.

Suppose Σ' is consistent and $\mathscr{B}$ is a sentence of $\mathscr{L}'$. By T10.8, Σ'' is a maximal, consistent, scapegoat set. We begin with a preliminary result, which connects arbitrary variable-free terms to our treatment of constants in the example above: for any variable-free term t_z and variable assignment d, $\mathsf{M}'_\mathsf{d}[t_z] = \mathsf{z}$.

Suppose t_z is a variable-free term and d is an arbitrary variable assignment. By induction on the number of function symbols in t_z, $\mathsf{M}'_\mathsf{d}[t_z] = \mathsf{z}$.

Basis: If t_z has no function symbols, then it is a constant. In this case, by construction, $\mathsf{M}'[t_z] = \mathsf{z}$; so by TA(c), $\mathsf{M}'_\mathsf{d}[t_z] = \mathsf{z}$.

Assp: For any i, $0 \le i < k$, if t_z has i function symbols, then $\mathsf{M}'_\mathsf{d}[t_z] = \mathsf{z}$.

[3] It is common to let U just be the set of variable-free terms in $\mathscr{L}'$, and the interpretation of a term be itself. There is nothing the matter with this. However, working with the integers emphasizes continuity with other models we have seen, and positions us for further results.

Show: If t_z has k function symbols, then $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{z}$.

If t_z has k function symbols, then it is of the form $h^n t_a \dots t_b$ for function symbol h^n and variable-free terms $t_a \dots t_b$ each with $< k$ function symbols. By TA(f), $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{M}'_{\mathsf{d}}[h^n t_a \dots t_b] = \mathsf{M}'[h^n]\langle \mathsf{M}'_{\mathsf{d}}[t_a] \dots \mathsf{M}'_{\mathsf{d}}[t_b]\rangle$; but by assumption, $\mathsf{M}'_{\mathsf{d}}[t_a] = \mathsf{a}$, and ... and $\mathsf{M}'_{\mathsf{d}}[t_b] = \mathsf{b}$; so $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{M}'[h^n]\langle \mathsf{a} \dots \mathsf{b}\rangle$. But since $t_z = h^n t_a \dots t_b$ is a variable-free term, by construction, $\langle\langle \mathsf{a} \dots \mathsf{b}\rangle, \mathsf{z}\rangle \in \mathsf{M}'[h^n]$; so we have $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{M}'[h^n]\langle \mathsf{a} \dots \mathsf{b}\rangle = \mathsf{z}$.

Indct: For any t_z, $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{z}$.

Given this, we are ready to show, by induction on the number of operators in $\mathcal{B}$, that $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$. Suppose $\mathcal{B}$ is a sentence.

Basis: If $\mathcal{B}$ is a sentence with no operators, then it is a sentence letter $\mathcal{S}$, or an atomic $\mathcal{R}^n t_a \dots t_b$ for relation symbol $\mathcal{R}^n$ and variable-free terms $t_a \dots t_b$. In the first case, by construction, $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$. In the second case, by TI, $\mathsf{M}'[\mathcal{R}^n t_a \dots t_b] = \mathsf{T}$ iff for arbitrary d, $\mathsf{M}'_{\mathsf{d}}[\mathcal{R}^n t_a \dots t_b] = \mathsf{S}$; by SF(r), iff $\langle \mathsf{M}'_{\mathsf{d}}[t_a] \dots \mathsf{M}'_{\mathsf{d}}[t_b]\rangle \in \mathsf{M}'[\mathcal{R}^n]$; since $t_a \dots t_b$ are variable-free terms, by the above result, iff $\langle \mathsf{a} \dots \mathsf{b}\rangle \in \mathsf{M}'[\mathcal{R}^n]$; by construction, iff $\Sigma'' \vdash \mathcal{R}^n t_a \dots t_b$. In either case, then, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Assp: For any i, $0 \le i < k$ if a sentence $\mathcal{B}$ has i operator symbols, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Show: If a sentence $\mathcal{B}$ has k operator symbols, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

If $\mathcal{B}$ has k operator symbols, then it is of the form, $\sim\mathcal{P}$, $\mathcal{P} \rightarrow \mathcal{Q}$ or $\forall x \mathcal{P}$, for variable x and $\mathcal{P}$ and $\mathcal{Q}$ with $< k$ operator symbols.

($\sim$) Suppose $\mathcal{B}$ is $\sim\mathcal{P}$. Homework. Hint: given T8.6, your reasoning may be very much as in the sentential case.

($\rightarrow$) Suppose $\mathcal{B}$ is $\mathcal{P} \rightarrow \mathcal{Q}$. Homework.

($\forall$) Suppose $\mathcal{B}$ is $\forall x \mathcal{P}$. Then since $\mathcal{B}$ is a sentence, x is the only variable that could be free in $\mathcal{P}$.

(i) Suppose $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ but $\Sigma'' \nvdash \mathcal{B}$; from the latter, $\Sigma'' \nvdash \forall x \mathcal{P}$; since Σ'' is maximal, $\Sigma'' \vdash \sim\forall x \mathcal{P}$; and since Σ'' is a scapegoat set, for some constant c, $\Sigma'' \vdash \sim\mathcal{P}^x_c$; so by consistency, $\Sigma'' \nvdash \mathcal{P}^x_c$; but $\mathcal{P}^x_c$ is a sentence; so by assumption, $\mathsf{M}'[\mathcal{P}^x_c] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{M}'_{\mathsf{d}}[\mathcal{P}^x_c] \neq \mathsf{S}$; but, where c is some t_a, by construction, $\mathsf{M}'[c] = \mathsf{a}$; so by TA(c), $\mathsf{M}'_{\mathsf{d}}[c] = \mathsf{a}$; so, since c is free for x in $\mathcal{P}$, by T10.2,

$\mathsf{M}'_{\mathsf{d}(x|\mathsf{a})}[\mathcal{P}] \neq \mathsf{S}$; so by SF($\forall$), $\mathsf{M}'_{\mathsf{d}}[\forall x\mathcal{P}] \neq \mathsf{S}$; so by TI, $\mathsf{M}'[\forall x\mathcal{P}] \neq \mathsf{T}$; and this is just to say, $\mathsf{M}'[\mathcal{B}] \neq \mathsf{T}$. But this is impossible; reject the assumption: if $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$, then $\Sigma'' \vdash \mathcal{B}$.

(ii) Suppose $\Sigma'' \vdash \mathcal{B}$ but $\mathsf{M}'[\mathcal{B}] \neq \mathsf{T}$; from the latter, $\mathsf{M}'[\forall x\mathcal{P}] \neq \mathsf{T}$; so by TI, there is some d such that $\mathsf{M}'_{\mathsf{d}}[\forall x\mathcal{P}] \neq \mathsf{S}$; so by SF($\forall$), there is some $\mathsf{a} \in \mathsf{U}$ such that $\mathsf{M}'_{\mathsf{d}(x|\mathsf{a})}[\mathcal{P}] \neq \mathsf{S}$; but for variable-free term t_a, by our above result, $\mathsf{M}'_{\mathsf{d}}[t_a] = \mathsf{a}$, and since t_a is variable-free, it is free for x in $\mathcal{P}$, so by T10.2, $\mathsf{M}'_{\mathsf{d}}[\mathcal{P}^x_{t_a}] \neq \mathsf{S}$; so by TI, $\mathsf{M}'[\mathcal{P}^x_{t_a}] \neq \mathsf{T}$; but $\mathcal{P}^x_{t_a}$ is a sentence; so by assumption, $\Sigma'' \nvdash \mathcal{P}^x_{t_a}$; so by the maximality of Σ'', $\Sigma'' \vdash \sim\mathcal{P}^x_{t_a}$; but t_a is free for x in $\mathcal{P}$, so by A4, $\vdash \forall x\mathcal{P} \rightarrow \mathcal{P}^x_{t_a}$; and by T3.13, $\vdash (\forall x\mathcal{P} \rightarrow \mathcal{P}^x_{t_a}) \rightarrow (\sim\mathcal{P}^x_{t_a} \rightarrow \sim\forall x\mathcal{P})$; so by a couple instances of MP, $\Sigma'' \vdash \sim\forall x\mathcal{P}$; so by the consistency of Σ'', $\Sigma'' \nvdash \forall x\mathcal{P}$; which is to say, $\Sigma'' \nvdash \mathcal{B}$. This is impossible; reject the assumption: if $\Sigma'' \vdash \mathcal{B}$, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$.

If $\mathcal{B}$ has k operator symbols, then $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Indct: For any sentence $\mathcal{B}$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

So if Σ' is consistent, then for any sentence $\mathcal{B}$ of $\mathcal{L}'$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$. We are now just one step away from ($\star$). It will be easy to see that $\mathsf{M}'[\Sigma'] = \mathsf{T}$, and so to reach the final result.

E10.19. Complete the $\sim$ and $\rightarrow$ cases to complete the demonstration of T10.9. You should set up the complete demonstration, but may refer to the text for cases completed there, as the text refers cases to homework.

10.3.5 Final Result

And now we are in a position to get the final result. This works just as before. First,

T10.10. If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($\star$)

Suppose Σ' is consistent, but $\mathsf{M}'[\Sigma'] \neq \mathsf{T}$. From the latter, there is some formula $\mathcal{B} \in \Sigma'$ such that $\mathsf{M}'[\mathcal{B}] \neq \mathsf{T}$. Since $\mathcal{B} \in \Sigma'$, by construction, $\mathcal{B} \in \Sigma''$, so $\Sigma'' \vdash \mathcal{B}$; so, where $\mathcal{B}^c$ is the universal closure of $\mathcal{B}$, by application of Gen* as necessary, $\Sigma'' \vdash \mathcal{B}^c$; so since Σ' is consistent, by T10.9, $\mathsf{M}'[\mathcal{B}^c] = \mathsf{T}$; so by applications of T7.6 as necessary, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$. This is impossible; reject the assumption: if Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

Notice that this result applies to arbitrary sets of *formulas*. We are able to bridge between formulas and sentences by T10.6 and Gen*. But now we have the ($\star$) that we have needed for adequacy.

So that is it! All we needed for the proof of adequacy was ($\star$). And we have it. So here is the final argument. Suppose the members of Γ and $\mathcal{P}$ are formulas of $\mathcal{L}'$.

T10.11. If $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. (*quantificational adequacy*)

Suppose $\Gamma \vDash \mathcal{P}$ but $\Gamma \nvdash \mathcal{P}$. Say, for the moment that $\Gamma \vdash \sim\sim\mathcal{P}^c$; by T3.10, $\vdash \sim\sim\mathcal{P}^c \rightarrow \mathcal{P}^c$; so by MP, $\Gamma \vdash \mathcal{P}^c$; so by repeated applications of A4 and MP, $\Gamma \vdash \mathcal{P}$; but this is impossible; so $\Gamma \nvdash \sim\sim\mathcal{P}^c$. Given this, since $\sim\sim\mathcal{P}^c$ is a sentence, by T10.6, $\Gamma \cup \{\sim\mathcal{P}^c\} = \Sigma'$ is consistent; so by T10.10, there is a model M' constructed as above such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$. So $\mathsf{M}'[\Gamma] = \mathsf{T}$ and $\mathsf{M}'[\sim\mathcal{P}^c] = \mathsf{T}$; from the latter, by T8.6, $\mathsf{M}'[\mathcal{P}^c] \neq \mathsf{T}$; so by repeated applications of T7.6, $\mathsf{M}'[\mathcal{P}] \neq \mathsf{T}$; so by QV, $\Gamma \nvDash \mathcal{P}$. This is impossible; reject the assumption: if $\Gamma \vDash \mathcal{P}$ then $\Gamma \vdash \mathcal{P}$.

Again, you should try to get the complete picture in your mind: The key is that consistent sets always have models. If $\Gamma \cup \{\sim\mathcal{P}\}$ is not consistent, then there is a derivation of $\mathcal{P}$ from Γ. So if there is no derivation of $\mathcal{P}$ from Γ, $\Gamma \cup \{\sim\mathcal{P}\}$ is consistent and so must have a model — with the result that $\Gamma \nvDash \mathcal{P}$. We get the key point, that consistent sets have models, by finding a relation between consistent, and *maximal*, consistent, scapegoat sets. If a set is maximal and consistent and a scapegoat set, then it contains enough information to specify a model for the whole. The model for the big set then guarantees the existence of a model M for the original Γ. All of this is very much parallel to the sentential case.

E10.20. Consider a quantificational language $\mathcal{L}$ which has function symbols as usual but with $\wedge$, $\sim$, and $\exists$ as primitive operators. Suppose axioms and rules are as in *A4* of E10.3 on p. 470. You may suppose there is no symbol for equality, and there are infinitely many constants not in Γ. Provide a complete demonstration that *A4* is adequate. You may appeal to any results from the text whose demonstration remains unchanged, but should recreate parts whose demonstration is not the same.

Hints: As preliminaries you will need revised versions of DT and T10.12. In addition, a few quick theorems for derivations, along with an analog to one side of T7.6 might be helpful,

(a) $\vdash \exists y \mathcal{P}^x_y \rightarrow \exists x \mathcal{P}$ $\qquad$ y free for x in $\mathcal{P}$ and not free in $\exists x \mathcal{P}$

(b) $\vdash \sim\exists x\mathcal{P} \to \sim\exists y\mathcal{P}^x_y$ $\quad$ y free for x in $\mathcal{P}$ and not free in $\exists x\mathcal{P}$

(c) $\sim\mathcal{P}^x_v \vdash \sim\exists x\mathcal{P}$ $\quad$ use $\exists$E with $\mathcal{Q}$ some $X \wedge \sim X$; note that $\models \sim(X \wedge \sim X)$

(7.6*) If $\mathsf{I}[\sim\exists x\mathcal{P}] = \mathsf{T}$ then $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$

Then redefine key notions (such as 'scapegoat set') in terms of the existential quantifier, so that you can work cases directly within the new system. Say $\mathcal{P}^e$ is the *existential* closure of $\mathcal{P}$. Note that $\sim(\sim\mathcal{P})^e$ is equivalent to $\mathcal{P}^c$ (imagine replacing all the added universal quantifiers in $\mathcal{P}^c$ with $\sim\exists x\sim$ and using DN on inner double tildes). This will help with T10.10 and T10.11.

10.4 Quantificational Adequacy: Full Version

So far, we have shown that if $\Gamma \models \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$ where the members of Γ and $\mathcal{P}$ are formulas of $\mathcal{L}'$. Now allow that the members of Γ and $\mathcal{P}$ are in an arbitrary quantificational language $\mathcal{L}$. Then we we shall require require not $(\star)$ with application just to $\mathcal{L}'$, but the more general,

($\star\star$) For any consistent set of formulas Σ, there is an interpretation M such that $\mathsf{M}[\Sigma] = \mathsf{T}$.

Given this, reasoning is exactly as before.

1.	$\Gamma \cup \{\sim\mathcal{P}^c\}$ has a model	$\Longrightarrow$	$\Gamma \not\models \mathcal{P}$	
2.	$\Gamma \cup \{\sim\mathcal{P}^c\}$ is consistent	$\Longrightarrow$	$\Gamma \cup \{\sim\mathcal{P}^c\}$ has a model	($\star\star$)
3.	$\Gamma \cup \{\sim\mathcal{P}^c\}$ is not consistent	$\Longrightarrow$	$\Gamma \vdash \mathcal{P}$	

Reasoning for (1) and (3) remains the same. (2) is ($\star\star$). Now suppose $\Gamma \models \mathcal{P}$; then from (1), $\Gamma \cup \{\sim\mathcal{P}^c\}$ does not have a model; so by (2), $\Gamma \cup \{\sim\mathcal{P}^c\}$ is not consistent; so by (3), $\Gamma \vdash \mathcal{P}$. So if $\Gamma \models \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. Supposing that ($\star\star$) has application to arbitrary sets of formulas, the result has application to arbitrary premises and conclusion. So we are left with two issues relative to our reasoning from before: $\mathcal{L}$ might lack the infinitely many constants not in the premises, and $\mathcal{L}$ might include equality.

10.4.1 Adding Constants

Suppose $\mathscr{L}$ does not have infinitely many constants not in Γ. This can happen in different ways. Perhaps $\mathscr{L}$ simply does not have infinitely many constants. Or perhaps the constants of $\mathscr{L}$ are a_1, $a_2 \ldots$ and $\Gamma = \{\mathcal{R}a_1, \mathcal{R}a_2 \ldots\}$; then $\mathscr{L}$ has infinitely many constants, but there are not any constants in $\mathscr{L}$ that do not appear in Γ. And we need the extra constants for construction of the maximal, consistent, scapegoat set. To avoid this sort of worry, we simply *add* infinitely many constants to form a language $\mathscr{L}'$ out of $\mathscr{L}$.

Cns$\mathscr{L}'$ Where $\mathscr{L}$ is a language whose constants are some of a_1, $a_2 \ldots$ let $\mathscr{L}'$ be like $\mathscr{L}$ but with the addition of new constants c_1, $c_2 \ldots$

By reasoning as in the countability reference on p. 33, insofar as they can be lined up, a_1, c_1, a_2, $c_2 \ldots$ the collection of constants remains countable, so that $\mathscr{L}'$ remains a perfectly legitimate quantificational language. Clearly, every formula of $\mathscr{L}$ remains a formula of $\mathscr{L}'$. Thus, where Σ is a set of formulas in language $\mathscr{L}$, let Σ' be like Σ except that its members are formulas of language $\mathscr{L}'$.

Our reasoning for $(\star)$ has application to sets of the sort Σ'. That is, where $\mathscr{L}'$ has infinitely many constants not in Σ', we have been able to find a maximal, consistent, scapegoat set Σ'', and from this a model M' for Σ'. But, give an arbitrary Σ of formulas in $\mathscr{L}$, we need that *it* has a model M. That is, we shall have to establish a bridge between Σ and Σ', and between M' and M. Thus, to obtain $(\star\star)$, we show,

2a.	Σ is consistent	$\Longrightarrow$	Σ' is consistent
2b.	Σ' is consistent	$\Longrightarrow$	Σ' has a model M'
2c.	Σ' has a model M'	$\Longrightarrow$	Σ has a model M

(2b) is just $(\star)$ from before. And by a sort of hypothethical syllogism, together these yield $(\star\star)$.

For the first result, we need that if Σ is consistent, then Σ' is consistent. Of course, Σ and Σ' contain just the same formulas, only sentences of the one are in a language with extra constants. But there might be *derivations* in $\mathscr{L}'$ from Σ' that are not derivations in $\mathscr{L}$ from Σ. So we need to show that these extra derivations do not result in contradiction. For this, the overall idea is simple: If we can derive a contradiction from Σ' in the enriched language then, by a modified version of that very derivation, we can derive a contradiction from Σ in the reduced language. So if there is no contradiction in the reduced language $\mathscr{L}$, then there can be no contradiction in the enriched language $\mathscr{L}'$. The argument is straightforward, given the preliminary

result T10.12. Let Σ be a set of formulas in $\mathscr{L}$, and Σ' those same formulas in $\mathscr{L}'$. We show,

T10.13. If Σ is consistent, then Σ' is consistent.

Suppose Σ is consistent. If Σ' is not consistent, then there is a formula $\mathcal{A}$ in $\mathscr{L}'$ such that $\Sigma' \vdash \mathcal{A}$ and $\Sigma' \vdash \sim\mathcal{A}$; but by T9.4, $\vdash \mathcal{A} \to [\sim\mathcal{A} \to (\mathcal{A} \wedge \sim\mathcal{A})]$; so by two instances of MP, $\Sigma' \vdash \mathcal{A} \wedge \sim\mathcal{A}$. So if Σ' is not consistent, there is a derivation of a contradiction from Σ'. By induction on the number of new constants which appear in a derivation $D = \langle \mathcal{B}_1, \mathcal{B}_2 \ldots \rangle$, we show that no such D is a derivation of a contradiction from Σ'.

Basis: Suppose D contains no new constants and D is a derivation of some contradiction $\mathcal{A} \wedge \sim\mathcal{A}$ from Σ'. Since D contains no new constants, every member of D is also a formula of $\mathscr{L}$, so $D = \langle \mathcal{B}_1, \mathcal{B}_2 \ldots \rangle$ is a derivation of $\mathcal{A} \wedge \sim\mathcal{A}$ from Σ; so by T3.19 and T3.20 with MP, $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \sim\mathcal{A}$; so Σ is not consistent. This is impossible; reject the assumption: D is not a derivation of a contradiction from Σ'.

Assp: For any i, $0 \le i < k$, if D contains i new constants, then it is not a derivation of a contradiction from Σ'.

Show: If D contains k new constants, then it is not a derivation of a contradiction from Σ'.

Suppose D contains k new constants and is a derivation of a contradiction $\mathcal{A} \wedge \sim\mathcal{A}$ from Σ'. Where c is one of the new constants in D and x is a variable not in D, by T10.12, D^{c}_{x} is a derivation of $[\mathcal{A} \wedge \sim\mathcal{A}]^{c}_{x}$ from Σ'^{c}_{x}. But all the members of Σ' are in $\mathscr{L}$; so c does not appear in any member of Σ'; so $\Sigma'^{c}_{x} = \Sigma'$. And $[\mathcal{A} \wedge \sim\mathcal{A}]^{c}_{x} = \mathcal{A}^{c}_{x} \wedge \sim[\mathcal{A}^{c}_{x}]$. So D^{c}_{x} is a derivation of a contradiction from Σ'. But D^{c}_{x} has $k-1$ new constants and so, by assumption, is not a derivation of a contradiction from Σ'. This is impossible; reject the assumption: D is not a derivation of a contradiction from Σ'.

Indct: No derivation D is a derivation of a contradiction from Σ'.

So if Σ is consistent, then Σ' is consistent. So if we have a consistent set of sentences in $\mathscr{L}$, and convert to $\mathscr{L}'$ with additional constants, we can be sure that the converted set is consistent as well.

With the extra constants in-hand, all our reasoning goes through as before to show that there is a model M' for Σ'. Officially, though, an interpretation for some

sentences in $\mathscr{L}'$ is not a model for some sentences in $\mathscr{L}$: a model for sentences in $\mathscr{L}$ has assignments for its constants, function symbols and relation symbols, where a model for $\mathscr{L}'$ has assignments for *its* constants, function symbols and relation symbols. A model M' for Σ', then, is not the same as a model M for Σ. But it is a short step to a solution.

CnsM Let M be like M' but without assignments to constants not in $\mathscr{L}$.

M is an interpretation for language $\mathscr{L}$. M and M' have exactly the same universe of discourse, and exactly the same interpretations for all the symbols that are in $\mathscr{L}$. It turns out that the evaluation of any formula in $\mathscr{L}$ is therefore the same on M as on M' — that is, for any $\mathscr{P}$ in $\mathscr{L}$, $\mathsf{M}[\mathscr{P}] = \mathsf{T}$ iff $\mathsf{M}'[\mathscr{P}] = \mathsf{T}$. Perhaps this is obvious. However, it is worthwhile to consider a proof. Thus we need the following matched pair of theorems (in fact, we show somewhat more than is necessary, as M and M' differ only by assignments to constants). The proofs are straightforward, and mostly left as an exercise. I do just enough to get you started.

Suppose $\mathscr{L}'$ extends $\mathscr{L}$ and M' is like M except that it makes assignments to constants, functions symbols and relation symbols in $\mathscr{L}'$ but not in $\mathscr{L}$.

T10.14. For any variable assignment d, and for any term t in $\mathscr{L}$, $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

The argument is by induction on the number of function symbols in t. Let d be a variable assignment, and t a term in $\mathscr{L}$.

Basis: Homework

Assp: For any i, $0 \le i < k$, if t has i function symbols, then $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

Show: If t has k function symbols, then $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

If t has k function symbols, then it is of the form, $h^n t_1 \ldots t_n$ for function symbol h^n and terms $t_1 \ldots t_n$ with $< k$ function symbols. By TA(f), $\mathsf{M_d}[t] = \mathsf{M_d}[h^n t_1 \ldots t_n] = \mathsf{M}[h^n]\langle \mathsf{M_d}[t_1] \ldots \mathsf{M_d}[t_n]\rangle$; similarly, $\mathsf{M'_d}[t] = \mathsf{M'_d}[h^n t_1 \ldots t_n] = \mathsf{M'}[h^n]\langle \mathsf{M'_d}[t_1] \ldots \mathsf{M'_d}[t_n]\rangle$. But by assumption, $\mathsf{M_d}[t_1] = \mathsf{M'_d}[t_1]$, and …and $\mathsf{M_d}[t_n] = \mathsf{M'_d}[t_n]$; and by construction, $\mathsf{M}[h^n] = \mathsf{M'}[h^n]$; so $\mathsf{M}[h^n]\langle \mathsf{M_d}[t_1] \ldots \mathsf{M_d}[t_n]\rangle = \mathsf{M'}[h^n]\langle \mathsf{M'_d}[t_1] \ldots \mathsf{M'_d}[t_n]\rangle$; so $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

Indct: For any t in $\mathscr{L}$, $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

T10.15. For any variable assignment d, and for any formula $\mathscr{P}$ in $\mathscr{L}$, $\mathsf{M_d}[\mathscr{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathscr{P}] = \mathsf{S}$.

The argument is by induction on the number of operator symbols in $\mathcal{P}$. Let d be a variable assignment, and $\mathcal{P}$ a formula in $\mathcal{L}$.

Basis: If $\mathcal{P}$ has no operator symbols, then it is a sentence letter $\mathcal{S}$ or an atomic $\mathcal{R}^n t_1 \dots t_n$ for relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$ in $\mathcal{L}$. In the first case, by SF(s), $\mathsf{M_d}[\mathcal{S}] = \mathsf{S}$ iff $\mathsf{M}[\mathcal{S}] = \mathsf{T}$; by construction, iff $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$; by SF(s), iff $\mathsf{M}'_\mathsf{d}[\mathcal{S}] = \mathsf{S}$. In the second case, by SF(r), $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$; iff $\langle \mathsf{M_d}[t_1] \dots \mathsf{M_d}[t_n]\rangle \in \mathsf{M}[\mathcal{R}^n]$; similarly, $\mathsf{M}'_\mathsf{d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M}'_\mathsf{d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$; iff $\langle \mathsf{M}'_\mathsf{d}[t_1] \dots \mathsf{M}'_\mathsf{d}[t_n]\rangle \in \mathsf{M}'[\mathcal{R}^n]$. But by T10.14, $\mathsf{M_d}[t_1] = \mathsf{M}'_\mathsf{d}[t_1]$, and ... and $\mathsf{M_d}[t_n] = \mathsf{M}'_\mathsf{d}[t_n]$; and by construction, $\mathsf{M}[\mathcal{R}^n] = \mathsf{M}'[\mathcal{R}^n]$; so $\langle \mathsf{M_d}[t_1] \dots \mathsf{M_d}[t_n]\rangle \in \mathsf{M}[\mathcal{R}^n]$ iff $\langle \mathsf{M}'_\mathsf{d}[t_1] \dots \mathsf{M}'_\mathsf{d}[t_n]\rangle \in \mathsf{M}'[\mathcal{R}^n]$; so $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M}'_\mathsf{d}[\mathcal{P}] = \mathsf{S}$.

Assp: For any i, $0 \le i < k$, and any variable assignment d, if $\mathcal{P}$ has i operator symbols, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M}'_\mathsf{d}[\mathcal{P}] = \mathsf{S}$.

Show: Homework

Indct: For any formula $\mathcal{P}$ of $\mathcal{L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M}'_\mathsf{d}[\mathcal{P}] = \mathsf{S}$.

And now we are in a position to show that M is indeed a model for Σ. In particular, it is easy to show,

T10.16. If $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

Suppose $\mathsf{M}'[\Sigma'] = \mathsf{T}$, but $\mathsf{M}[\Sigma] \neq \mathsf{T}$. From the latter, there is some formula $\mathcal{B} \in \Sigma$ such that $\mathsf{M}[\mathcal{B}] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{M_d}[\mathcal{B}] \neq \mathsf{S}$; so by T10.15, $\mathsf{M}'_\mathsf{d}[\mathcal{B}] \neq \mathsf{S}$; so by TI, $\mathsf{M}'[\mathcal{B}] \neq \mathsf{T}$; and since $\mathcal{B} \in \Sigma$, we have $\mathcal{B} \in \Sigma'$; so $\mathsf{M}'[\Sigma'] \neq \mathsf{T}$. This is impossible; reject the assumption: if $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

T10.13, T10.10, and T10.16 together yield,

T10.17. $\mathcal{L}$, if Σ is consistent, then Σ has a model M ($\mathcal{L}$ without equality).

Suppose Σ is consistent; then by T10.13, Σ' is consistent; so by T10.10, Σ' has a model M'; so by T10.16, Σ has a model M.

And that is what we needed to recover the adequacy result for $\mathcal{L}$ without the constraint on constants. Where $\mathcal{L}$ does not include infinitely many constants not in Γ, we simply add them to form $\mathcal{L}'$. Our theorems from this section ensure that the results go through as before.

*E10.21. Complete the proof of T10.14. You should set up the complete induction, but may refer to the text, as the text refers to homework.

*E10.22. Complete the proof of T10.15. As usual, you should set up the complete induction, but may refer to the text for cases completed there, as the text refers to homework.

E10.23. Adapt the demonstration of T10.11 for the supposition that $\mathscr{L}$ need not be the same as $\mathscr{L}'$. You may appeal to theorems from this section.

10.4.2 Accommodating Equality

Dropping the assumption that language $\mathscr{L}$ lacks the symbol '$=$' for equality results in another sort of complication. In constructing our models, where t_1 and t_3 from the enumeration of variable-free terms are constants and $\Sigma'' \vdash \mathscr{R}t_1t_3$, we set $\mathsf{M}'[t_1] = 1$, $\mathsf{M}'[t_3] = 3$ and $\langle 1, 3\rangle \in \mathsf{M}'[\mathscr{R}]$. But suppose $\mathscr{R}$ is the equal sign, '$=$'; then by our procedure, $\langle 1, 3\rangle \in \mathsf{M}'[=]$. But this is wrong! Where $\mathsf{U} = \{1, 2 \ldots\}$, the proper interpretation of '$=$' is $\{\langle 1, 1\rangle, \langle 2, 2\rangle \ldots\}$, and $\langle 1, 3\rangle$ is not a member of this set at all. So our procedure does not result in the specification of a legitimate model. The procedure works fine for relation symbols other than equality. There are no restrictions on assignments to other relation symbols, so nothing stops us from specifying interpretations as above. But there is a restriction on the interpretation of '$=$'. So we cannot proceed blindly this way.

Here is the nub of a solution: Say $\Sigma'' \vdash a_1 = a_3$; then let the *set* $\{1, 3\}$ be an element of U, and let $\mathsf{M}'[a_1] = \mathsf{M}'[a_3] = \{1, 3\}$. Similarly, if $a_2 = a_4$ and $a_4 = a_5$ are consequences of Σ'', let $\{2, 4, 5\}$ be a member of U, and $\mathsf{M}'[a_2] = \mathsf{M}'[a_4] = \mathsf{M}'[a_5] = \{2, 4, 5\}$. That is, let U consist of certain *sets* of integers — where these sets are specified by atomic equalities that are consequences of Σ''. Then let $\mathsf{M}'[a_z]$ be the set of which z is a member. Given this, if $\Sigma'' \vdash \mathscr{R}^n t_a \ldots t_b$, then include the tuple consisting of the set assigned to t_a, and ... and the set assigned to t_b, in the interpretation of $\mathscr{R}^n$. So on the above interpretation of the constants, if $\Sigma'' \vdash \mathscr{R}a_1a_4$, then $\langle\{1, 3\}, \{2, 4, 5\}\rangle \in \mathsf{M}'[\mathscr{R}]$. And if $\Sigma'' \vdash a_1 = a_3$, then $\langle\{1, 3\}, \{1, 3\}\rangle \in \mathsf{M}'[=]$. You should see why this is so. And it is just right! If $\{1, 3\} \in \mathsf{U}$, then $\langle\{1, 3\}, \{1, 3\}\rangle$ *should* be in $\mathsf{M}'[=]$. So we respond to the problem by a revision of the specification for CnsM$'$.

Let us now turn to the details. Put abstractly, the reason the argument in the basis of T10.9 works is that our model M' assigns each t in the enumeration of variable-free terms an object m such that whenever $\Sigma'' \vdash \mathscr{R}t$ then $\mathsf{m} \in \mathsf{M}'[\mathscr{R}]$; and for the

universal case, it is important that for each object there is a constant to which it is assigned. We want an interpretation that preserves these features. And it will be important to demonstrate that our specifications are coherent. A model consists of a universe U, along with assignments to constants, function symbols, sentence letters, and relation symbols. We take up these elements, one after another.

The universe. The elements of our universe U are to be certain sets of integers.[4] Consider an enumeration $t_1, t_2 \ldots$ of all the variable-free terms in $\mathscr{L}'$, and let there be a relation $\simeq$ on the set $\{1, 2 \ldots\}$ of positive integers such that $\mathsf{i} \simeq \mathsf{j}$ iff $\Sigma'' \vdash t_i = t_j$. Let $\overline{\mathsf{n}}$ be the set of integers which stand in the $\simeq$ relation to n — that is, $\overline{\mathsf{n}} = \{\mathsf{z} \mid \mathsf{z} \simeq \mathsf{n}\}$. So whenever $\mathsf{z} \simeq \mathsf{n}$, then $\mathsf{z} \in \overline{\mathsf{n}}$. The universe U of M' is then the collection of all these sets — that is,

CnsM$'$ For each integer greater than or equal to one, the universe includes the class corresponding to it. $\mathsf{U} = \{\overline{\mathsf{n}} \mid \mathsf{n} \geq 1\}$.

The way this works is really quite simple. If according to Σ'', t_1 equals only itself, then the only z such that $\mathsf{z} \simeq 1$ is 1; so $\overline{1} = \{1\}$, and this is a member of U. If, according to Σ'', t_1 equals just itself and t_2, then $1 \simeq 2$ so that $\overline{1} = \overline{2} = \{1, 2\}$, and this set is a member of U. If, according to Σ'', t_1 equals itself, t_2 and t_3, then $1 \simeq 2 \simeq 3$ so that $\overline{1} = \overline{2} = \overline{3} = \{1, 2, 3\}$, and this set is a member of U. And so forth.

In order to make progress, it will be convenient to establish some facts about the $\simeq$ relation, and about the sets in U. Recall that $\simeq$ is a relation on the *integers* which is specified relative to expressions in Σ'', so that $\mathsf{i} \simeq \mathsf{j}$ iff $\Sigma'' \vdash t_i = t_j$. First we show that $\simeq$ is *reflexive*, *symmetric*, and *transitive*.

> *Reflexivity.* For any i, $\mathsf{i} \simeq \mathsf{i}$. By T3.32, $\vdash t_i = t_i$; so $\Sigma'' \vdash t_i = t_i$; so by construction, $\mathsf{i} \simeq \mathsf{i}$.
>
> *Symmetry.* For any i and j, if $\mathsf{i} \simeq \mathsf{j}$, then $\mathsf{j} \simeq \mathsf{i}$. Suppose $\mathsf{i} \simeq \mathsf{j}$; then by construction, $\Sigma'' \vdash t_i = t_j$; but by T3.33, $\vdash t_i = t_j \rightarrow t_j = t_i$; so by MP, $\Sigma'' \vdash t_j = t_i$; so by construction, $\mathsf{j} \simeq \mathsf{i}$.
>
> *Transitivity.* For any i, j and k, if $\mathsf{i} \simeq \mathsf{j}$ and $\mathsf{j} \simeq \mathsf{k}$, then $\mathsf{i} \simeq \mathsf{k}$. Suppose $\mathsf{i} \simeq \mathsf{j}$ and $\mathsf{j} \simeq \mathsf{k}$; then by construction, $\Sigma'' \vdash t_i = t_j$ and $\Sigma'' \vdash t_j = t_k$; but by

[4] Again, it is common to let the universe be *sets of terms* in $\mathscr{L}'$. There is nothing the matter with this. However, working with the integers emphasizes continuity with other models we have seen, and positions us for further results.

T3.34, $\vdash t_i = t_j \to (t_j = t_k \to t_i = t_k)$; so by two instances of MP, $\Sigma'' \vdash t_i = t_k$; so by construction, $\mathsf{i} \simeq \mathsf{k}$.

A relation which is reflexive, symmetric and transitive is called an *equivalence* relation. As an equivalence relation, it divides or *partitions* the members of $\{1, 2 \ldots\}$ into mutually exclusive classes such that each member of a class bears $\simeq$ to each of the others in its partition, but not to integers outside the partition. More particularly, because $\simeq$ is an equivalence relation, the collections $\overline{\mathsf{n}} = \{\mathsf{z} \mid \mathsf{z} \simeq \mathsf{n}\}$ in U are characterized as follows.

Self-membership. For any n, $\mathsf{n} \in \overline{\mathsf{n}}$. By reflexivity, $\mathsf{n} \simeq \mathsf{n}$; so by construction, $\mathsf{n} \in \overline{\mathsf{n}}$. Corollary: Every integer i is a member of at least one class.

Uniqueness. For any i, i is an an element of at most one class. Suppose i is an element of more than one class; then there are some m and n such that $\mathsf{i} \in \overline{\mathsf{m}}$ and $\mathsf{i} \in \overline{\mathsf{n}}$ but $\overline{\mathsf{m}} \neq \overline{\mathsf{n}}$. Since $\overline{\mathsf{m}} \neq \overline{\mathsf{n}}$ there is some j such that $\mathsf{j} \in \overline{\mathsf{m}}$ and $\mathsf{j} \notin \overline{\mathsf{n}}$, or $\mathsf{j} \in \overline{\mathsf{n}}$ and $\mathsf{j} \notin \overline{\mathsf{m}}$; without loss of generality, suppose $\mathsf{j} \in \overline{\mathsf{m}}$ and $\mathsf{j} \notin \overline{\mathsf{n}}$. Since $\mathsf{j} \in \overline{\mathsf{m}}$, by construction, $\mathsf{j} \simeq \mathsf{m}$; and since $\mathsf{i} \in \overline{\mathsf{m}}$, by construction $\mathsf{i} \simeq \mathsf{m}$; so by symmetry, $\mathsf{m} \simeq \mathsf{i}$; so by transitivity, $\mathsf{j} \simeq \mathsf{i}$. Since $\mathsf{i} \in \overline{\mathsf{n}}$, by construction $\mathsf{i} \simeq \mathsf{n}$; so by transitivity again, $\mathsf{j} \simeq \mathsf{n}$; so by construction, $\mathsf{j} \in \overline{\mathsf{n}}$. This is impossible; reject the assumption: i is an element of at most one class.

Equality. For any m and n, $\mathsf{m} \simeq \mathsf{n}$ iff $\overline{\mathsf{m}} = \overline{\mathsf{n}}$. (i) Suppose $\mathsf{m} \simeq \mathsf{n}$. Then by construction, $\mathsf{m} \in \overline{\mathsf{n}}$; but by self-membership, $\mathsf{m} \in \overline{\mathsf{m}}$; so by uniqueness, $\overline{\mathsf{n}} = \overline{\mathsf{m}}$. Suppose $\overline{\mathsf{m}} = \overline{\mathsf{n}}$; by self-membership, $\mathsf{m} \in \overline{\mathsf{m}}$; so $\mathsf{m} \in \overline{\mathsf{n}}$; so by construction, $\mathsf{m} \simeq \mathsf{n}$.

Corresponding to the relations by which they are formed, classes characterized by self-membership, uniqueness and equality are *equivalence classes*. From self-membership and uniqueness, every n is a member of exactly one such class. And from equality, $\mathsf{m} \simeq \mathsf{n}$ just when $\overline{\mathsf{m}}$ is the very same thing as $\overline{\mathsf{n}}$. So, for example, if $1 \simeq 1$ and $2 \simeq 1$ (and nothing else), then $\overline{1} = \overline{2} = \{1, 2\}$. You should be able to see that these formal specifications develop just the informal picture with which we began.

Terms. The specification for constants is simple.

CnsM′ If t_z in the enumeration of variable-free terms t_1, $t_2 \ldots$ is a constant, then $\mathsf{M}'[t_z] = \overline{\mathsf{z}}$.

Thus, with self-membership, any constant t_z designates the equivalence class of which z is a member. In this case, we need to be sure that the specification picks out exactly one member of U for each constant. The specification would fail if the relation $\simeq$ generated classes such that some integer was an element of no class, or some integer was an element of more than one. But, as we have just seen, by self-membership and uniqueness, every z is a member of exactly one class. So far, so good!

CnsM$'$ If t_z in the enumeration of variable-free terms $t_1,\ t_2 \ldots$ is $h^n t_a \ldots t_b$ for function symbol h^n and variable-free terms $t_a \ldots t_b$, then $\langle\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{b}}\rangle, \overline{\mathsf{z}}\rangle \in \mathsf{M}'[h^n]$.

Thus when the input to h^n is $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{b}}\rangle$, the output is $\overline{\mathsf{z}}$. This time, we must be sure that the result is a function — that (i) there is a defined output object for every input n-tuple, and (ii) there is at most one output object associated with any one input n-tuple. The former worry is easily dispatched. The second concern is that there might be some $t_m = h t_a$ and $t_n = h t_b$ in the list of variable-free terms, where $\overline{\mathsf{a}} = \overline{\mathsf{b}}$. Then $\langle \overline{\mathsf{a}}, \overline{\mathsf{m}}\rangle$, $\langle \overline{\mathsf{b}}, \overline{\mathsf{n}}\rangle \in \mathsf{M}'[h]$, and we fail to specify a function.

> (i) There is at least one output object. Corresponding to any $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{b}}\rangle$ where $\overline{\mathsf{a}} \ldots \overline{\mathsf{b}}$ are members of U, there is some variable-free $t_z = h^n t_a \ldots t_b$ in the sequence $t_1,\ t_2 \ldots$; so by construction, $\langle\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{b}}\rangle, \overline{\mathsf{z}}\rangle \in \mathsf{M}'[h^n]$. So $\mathsf{M}'[h^n]$ has a defined output object when the input is $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{b}}\rangle$.
>
> (ii) There is at most one output object. Suppose $\langle\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}}\rangle, \overline{\mathsf{m}}\rangle \in \mathsf{M}'[h^n]$ and $\langle\langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}}\rangle, \overline{\mathsf{n}}\rangle \in \mathsf{M}'[h^n]$, where $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}}\rangle = \langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}}\rangle$, but $\overline{\mathsf{m}} \neq \overline{\mathsf{n}}$. Since $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}}\rangle = \langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}}\rangle$, $\overline{\mathsf{a}} = \overline{\mathsf{d}}$, and ... and $\overline{\mathsf{c}} = \overline{\mathsf{f}}$; so by equality, $\mathsf{a} \simeq \mathsf{d}$, and ... and $\mathsf{c} \simeq \mathsf{f}$; so by construction, $\Sigma'' \vdash t_a = t_d$, and ... and $\Sigma'' \vdash t_c = t_f$. Since $\langle\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}}\rangle, \overline{\mathsf{m}}\rangle \in \mathsf{M}'[h^n]$ and $\langle\langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}}\rangle, \overline{\mathsf{n}}\rangle \in \mathsf{M}'[h^n]$, by construction, there are some variable-free terms, $t_m = h^n t_a \ldots t_c$ and $t_n = h^n t_d \ldots t_f$ in the enumeration; but by T3.36, $\vdash t_b = t_e \to h^n t_a \ldots t_b \ldots t_c = h^n t_a \ldots t_e \ldots t_c$, and so forth; so collecting repeated applications of this theorem with MP and T3.35, $\Sigma'' \vdash h^n t_a \ldots t_c = h^n t_d \ldots t_f$; but this is to say, $\Sigma'' \vdash t_m = t_n$; so by construction, $\mathsf{m} \simeq \mathsf{n}$; so by equality, $\overline{\mathsf{m}} = \overline{\mathsf{n}}$. This is impossible; reject the assumption: if $\langle\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}}\rangle, \overline{\mathsf{m}}\rangle \in \mathsf{M}'[h^n]$ and $\langle\langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}}\rangle, \overline{\mathsf{n}}\rangle \in \mathsf{M}'[h^n]$, where $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}}\rangle = \langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}}\rangle$, then $\overline{\mathsf{m}} = \overline{\mathsf{n}}$.

So, as they should be, functions are well-defined.

We are now in a position to recover an analogue to the preliminary result for demonstration of T10.9: for any variable-free term t_z and variable assignment d, $\mathsf{M}'_{\mathsf{d}}[t_z] = \overline{\mathsf{z}}$. The argument is very much as before. Suppose t_z is a variable-free term. By induction on the number of function symbols in t_z.

Basis: If t_z has no function symbols, then it is a constant. In this case, by construction, $\mathsf{M}'[t_z] = \overline{\mathsf{z}}$; so by TA(c), $\mathsf{M}'_{\mathsf{d}}[t_z] = \overline{\mathsf{z}}$.

Assp: For any i, $0 \le i < k$, if t_z has i function symbols, then $\mathsf{M}'_{\mathsf{d}}[t_z] = \overline{\mathsf{z}}$.

Show: If t_z has k function symbols, then $\mathsf{M}'_{\mathsf{d}}[t_z] = \overline{\mathsf{z}}$.

If t_z has k function symbols, then it is of the form, $\mathscr{h}^n t_a \dots t_b$ where $t_a \dots t_b$ have $< k$ function symbols. By TA(f) we have, $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{M}'_{\mathsf{d}}[\mathscr{h}^n t_a \dots t_b] = \mathsf{M}'[\mathscr{h}^n]\langle \mathsf{M}'_{\mathsf{d}}[t_a] \dots \mathsf{M}'_{\mathsf{d}}[t_b]\rangle$; but by assumption, $\mathsf{M}'_{\mathsf{d}}[t_a] = \overline{\mathsf{a}}$, and ... and $\mathsf{M}'_{\mathsf{d}}[t_b] = \overline{\mathsf{b}}$; so $\mathsf{M}'_{\mathsf{d}}[t_z] = \mathsf{M}'[\mathscr{h}^n]\langle \overline{\mathsf{a}} \dots \overline{\mathsf{b}}\rangle$. But since $t_z = \mathscr{h}^n t_a \dots t_b$ is a variable-free term, $\langle\langle \overline{\mathsf{a}} \dots \overline{\mathsf{b}}\rangle, \overline{\mathsf{z}}\rangle \in \mathsf{M}'[\mathscr{h}^n]$; so $\mathsf{M}'[\mathscr{h}^n]\langle \overline{\mathsf{a}} \dots \overline{\mathsf{b}}\rangle = \overline{\mathsf{z}}$; so $\mathsf{M}'_{\mathsf{d}}[t_z] = \overline{\mathsf{z}}$.

Indct: For any variable-free term t_z, $\mathsf{M}'_{\mathsf{d}}[t_z] = \overline{\mathsf{z}}$.

So the interepretation of any variable-free term is the equivalence class corresponding to its position in the enumeration of terms.

Atomics. The result we have just seen for terms makes the specification for atomics seem particularly natural. Sentence letters are easy. As before,

CnsM$'$ For a sentence letter $\mathscr{S}$, $\mathsf{M}'[\mathscr{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathscr{S}$.

Then for relation symbols, the idea is as sketched above. We simply let the assignment be such as to make a variable-free atomic come out true iff it is a consequence of Σ''.

CnsM$'$ For a relation symbol $\mathscr{R}^n$, where $t_a \dots t_b$ are n members of the enumeration of variable-free terms, let $\langle \overline{\mathsf{a}} \dots \overline{\mathsf{b}}\rangle \in \mathsf{M}'[\mathscr{R}^n]$ iff $\Sigma'' \vdash \mathscr{R}^n t_a \dots t_b$.

To see that the specification for relation symbols is legitimate, we need to be clear that the specification is consistent — that we do not both assert and deny that some tuple is in the extension of $\mathscr{R}^n$, and we need to be sure that $\mathsf{M}'[=]$ is as it should be — that it is $\{\langle \overline{\mathsf{n}}, \overline{\mathsf{n}}\rangle \mid \overline{\mathsf{n}} \in \mathsf{U}\}$. The case for equality is easy. The former concern is that we might have some $\overline{\mathsf{a}} \in \mathsf{M}'[\mathscr{R}]$ and $\overline{\mathsf{b}} \notin \mathsf{M}'[\mathscr{R}]$ but $\overline{\mathsf{a}} = \overline{\mathsf{b}}$.

(i) The specification is consistent. Suppose otherwise. Then there is some $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}} \rangle \in \mathsf{M}'[\mathcal{R}^n]$ and $\langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}} \rangle \notin \mathsf{M}'[\mathcal{R}^n]$, where $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}} \rangle = \langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}} \rangle$. From the latter, $\overline{\mathsf{a}} = \overline{\mathsf{d}}$, and ... and $\overline{\mathsf{c}} = \overline{\mathsf{f}}$; so by equality, $\mathsf{a} \simeq \mathsf{d}$, and ... and $\mathsf{c} \simeq \mathsf{f}$; so by construction, $\Sigma'' \vdash t_a = t_d$, and ... and $\Sigma'' \vdash t_c = t_f$. But since $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}} \rangle \in \mathsf{M}'[\mathcal{R}^n]$ and $\langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}} \rangle \notin \mathsf{M}'[\mathcal{R}^n]$, by construction, $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_c$ and $\Sigma'' \nvdash \mathcal{R}^n t_d \ldots t_f$; and by T3.37, $\vdash t_b = t_e \rightarrow (\mathcal{R}^n t_a \ldots t_b \ldots t_c \rightarrow \mathcal{R}^n t_a \ldots t_e \ldots t_c)$, and so forth; so by repeated applications of this theorem with MP, $\Sigma'' \vdash \mathcal{R}^n t_d \ldots t_f$. This is impossible; reject the assumption: if $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}} \rangle \in \mathsf{M}'[\mathcal{R}^n]$ and $\langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}} \rangle \notin \mathsf{M}'[\mathcal{R}^n]$, then $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{c}} \rangle \neq \langle \overline{\mathsf{d}} \ldots \overline{\mathsf{f}} \rangle$.

(ii) The case for equality is easy. By equality, $\overline{\mathsf{m}} = \overline{\mathsf{n}}$ iff $\mathsf{m} \simeq \mathsf{n}$; by construction iff $\Sigma'' \vdash t_m = t_n$; by construction iff $\langle \overline{\mathsf{m}}, \overline{\mathsf{n}} \rangle \in \mathsf{M}'[=]$.

This completes the specification of M'. The specification is more complex than for the basic version, and we have had to work to demonstrate its consistency. Still, the result is a perfectly ordinary model M', with a domain, assignments to constants, assignments to function symbols, and assignments to relation symbols.

With this revised specification for M', the demonstration of T10.9 proceeds as before. Here is the key portion of the basis. We are showing that $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

Suppose $\mathcal{B}$ is an atomic $\mathcal{R}^n t_a \ldots t_b$; then by TI, $\mathsf{M}'[\mathcal{R}^n t_a \ldots t_b] = \mathsf{T}$ iff for arbitrary d, $\mathsf{M}'_{\mathsf{d}}[\mathcal{R}^n t_a \ldots t_b] = \mathsf{S}$; by SF(r), iff $\langle \mathsf{M}'_{\mathsf{d}}[t_a] \ldots \mathsf{M}'_{\mathsf{d}}[t_b] \rangle \in \mathsf{M}'[\mathcal{R}^n]$; since $t_a \ldots t_b$ are variable-free terms, as we have just seen, iff $\langle \overline{\mathsf{a}} \ldots \overline{\mathsf{b}} \rangle \in \mathsf{M}'[\mathcal{R}^n]$; by construction, iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$. So $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

So all that happens is that we depend on the conversion from individuals to sets of individuals for both assignments to terms, and assignments to relation symbols. Given this, the argument is exactly parallel to the one from before.

E10.24. Suppose the enumeration of variable-free terms begins, a, b, $f^1 a$, $f^1 b \ldots$ (so these are $t_1 \ldots t_4$) and, for these terms, $\Sigma'' \vdash$ just $a = a$, $b = b$, $f^1 a = f^1 a$, $f^1 b = f^1 b$, $a = f^1 a$, and $f^1 a = a$. What objects stand in the $\simeq$ relation? What are $\overline{1}$, $\overline{2}$, $\overline{3}$, and $\overline{4}$? Which corresponding sets are members of U?

E10.25. Return to the case from E10.24. Explain how $\simeq$ satisfies reflexivity, symmetry and transitivity. Explain how U satisfies self-membership, uniqueness and equality.

E10.26. Where Σ'' and U are as in the previous two exercises, what are $\mathsf{M}'[a]$, $\mathsf{M}'[b]$ and $\mathsf{M}'[f]$? Supposing that $\Sigma'' \vdash R^1a$, R^1f^1a and R^1f^1b, but $\Sigma'' \nvdash R^1b$, what is $\mathsf{M}'[R^1]$? According to the method, what is $\mathsf{M}'[=]$? Is this as it should be? Explain.

10.4.3 The Final Result

We are really done with the demonstration of adequacy. Perhaps, though, it will be helpful to draw some parts together. Begin with the basic definitions.

Con A set Σ of formulas is *consistent* iff there is no formula $\mathcal{A}$ such that $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \sim\mathcal{A}$.

Max A set Σ of formulas is *maximal* iff for any sentence $\mathcal{A}$, $\Sigma \vdash \mathcal{A}$ or $\Sigma \vdash \sim\mathcal{A}$.

Scgt A set Σ of formulas is a *scapegoat* set iff for any sentence $\sim\forall x\mathcal{P}$, if $\Sigma \vdash \sim\forall x\mathcal{P}$, then there is some constant a such that $\Sigma \vdash \sim\mathcal{P}^x_a$.

Then we proceed in language $\mathcal{L}'$, for a maximal, consistent, scapegoat set Σ'' constructed from any consistent Σ'.

T10.6 For any set of formulas Σ and sentence $\mathcal{P}$, if $\Sigma \nvdash \sim\mathcal{P}$, then $\Sigma \cup \{\mathcal{P}\}$ is consistent.

T10.7 There is an enumeration $\mathcal{Q}_1$, $\mathcal{Q}_2 \ldots$ of all the formulas, terms, and the like, in $\mathcal{L}'$.

CnsΣ'' Construct Σ'' from Σ' as follows: By T10.7, there is an enumeration, $\mathcal{Q}_1$, $\mathcal{Q}_2 \ldots$ of all the sentences in $\mathcal{L}'$ and also an enumeration c_1, $c_2 \ldots$ of constants not in Σ'. Let $\Omega_0 = \Sigma'$. Then for any $i > 0$, let $\Omega_i = \Omega_{i-1}$ if $\Omega_{i-1} \vdash \sim\mathcal{Q}_i$. Otherwise, $\Omega_{i*} = \Omega_{i-1} \cup \{\mathcal{Q}_i\}$ if $\Omega_{i-1} \nvdash \sim\mathcal{Q}_i$. Then $\Omega_i = \Omega_{i*}$ if $\mathcal{Q}_i$ is not of the form $\sim\forall x\mathcal{P}$, and $\Omega_i = \Omega_{i*} \cup \{\sim\mathcal{P}^x_c\}$ if $\mathcal{Q}_i$ is of the form $\sim\forall x\mathcal{P}$, where c is the first constant not in Ω_{i*}. Then $\Sigma'' = \bigcup_{i\geq 0} \Omega_i$.

T10.8 If Σ' is consistent, then Σ'' is a maximal, consistent, scapegoat set.

Given the maximal, consistent, scapegoat set Σ'', there are results and a definition for a model M' such that $\mathsf{M}'[\Sigma'] = \mathsf{T}$.

CnsM′ $\mathsf{U} = \{\bar{\mathsf{n}} \mid \mathsf{n} \geq 1\}$. If t_z in an enumeration of variable-free terms $t_1, t_2 \ldots$ is a constant, then $\mathsf{M}'[t_z] = \bar{\mathsf{z}}$. If t_z is $h^n t_a \ldots t_b$ for function symbol h^n and variable-free terms $t_a \ldots t_b$, then $\langle\langle \bar{\mathsf{a}} \ldots \bar{\mathsf{b}}\rangle, \bar{\mathsf{z}}\rangle \in \mathsf{M}'[h^n]$. For a sentence letter $\mathcal{S}$, $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$. For a relation symbol $\mathcal{R}^n$, where $t_a \ldots t_b$ are n members of the enumeration of variable-free terms, let $\langle \bar{\mathsf{a}} \ldots \bar{\mathsf{b}}\rangle \in \mathsf{M}'[\mathcal{R}^n]$ iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$.

This modifies the relatively simple version where $\mathsf{U} = \{1, 2 \ldots\}$. And for an enumeration of variable-free terms, if t_z is a constant, $\mathsf{M}'[t_z] = \mathsf{z}$. If $t_z = h^n t_a \ldots t_b$ for some relation symbol h^n and n variable-free terms $t_a \ldots t_b$, $\langle\langle \mathsf{a} \ldots \mathsf{b}\rangle, \mathsf{z}\rangle \in \mathsf{M}'[h^n]$. For a sentence letter $\mathcal{S}$, $\mathsf{M}'[\mathcal{S}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{S}$. And for a relation symbol $\mathcal{R}^n$, $\langle \mathsf{a} \ldots \mathsf{b}\rangle \in \mathsf{M}'[\mathcal{R}^n]$ iff $\Sigma'' \vdash \mathcal{R}^n t_a \ldots t_b$.

T10.9 If Σ' is consistent, then for any sentence $\mathcal{B}$ of $\mathcal{L}'$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

T10.10 If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($\star$)

Then we have had to connect results for Σ' in $\mathcal{L}'$ to an arbitrary Σ in language $\mathcal{L}$.

T10.13 If Σ is consistent, then Σ' is consistent.

This is supported by T10.12 on which if D is a derivation from Σ', and x is a variable that does not appear in D, then for any constant a, D^a_x is a derivation from $\Sigma'\,{}^a_x$.

T10.16 If $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

This is supported by the matched pair of theorems, T10.14 on which, if d is a variable assignment, then for any term t in $\mathcal{L}$, $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$, and T10.15 on which, if d is a variable assignment, then for any formula $\mathcal{P}$ in $\mathcal{L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$.

These theorems together yield,

T10.17. If Σ is consistent, then Σ has a model M. ($\mathcal{L}$ unconstrained) ($\star\star$)

This puts us in a position to recover the main result. Recall that our argument runs through $\mathcal{P}^c$ the universal closure of $\mathcal{P}$.

T10.11. If $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. (*quantificational adequacy*)

Suppose $\Gamma \vDash \mathcal{P}$ but $\Gamma \nvdash \mathcal{P}$. Say, for the moment that $\Gamma \vdash {\sim}{\sim}\mathcal{P}^c$; by T3.10, $\vdash {\sim}{\sim}\mathcal{P}^c \to \mathcal{P}^c$; so by MP, $\Gamma \vdash \mathcal{P}^c$; so by repeated applications

of A4 and MP, $\Gamma \vdash \mathcal{P}$; but this is impossible; so $\Gamma \nvdash {\sim}{\sim}\mathcal{P}^c$. Given this, since ${\sim}{\sim}\mathcal{P}^c$ is a sentence, by T10.6, $\Gamma \cup \{{\sim}\mathcal{P}^c\}$ is consistent. Since $\Sigma = \Gamma \cup \{{\sim}\mathcal{P}^c\}$ is consistent, by T10.17, there is a model M constructed as above such that $\mathsf{M}[\Sigma] = \mathsf{T}$. So $\mathsf{M}[\Gamma] = \mathsf{T}$ and $\mathsf{M}[{\sim}\mathcal{P}^c] = \mathsf{T}$; from the latter, by T8.6, $\mathsf{M}[\mathcal{P}^c] \neq \mathsf{T}$; so by repeated applications of T7.6, $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; so by QV, $\Gamma \nvDash \mathcal{P}$. This is impossible; reject the assumption: if $\Gamma \vDash \mathcal{P}$ then $\Gamma \vdash \mathcal{P}$.

The sentential version had parallels to Con, Max, CnsΣ'' and CnsM' along with theorems T10.6_s - T10.11_s. (The distinction between ($\star$) and ($\star\star$) is a distinction without a difference in the sentential case.) The basic quantificational version requires these along with Sgt, T10.12 and the simple version of CnsM'. For the full version, we have had to appeal also to T10.13 and T10.16 (and so T10.17), and use the relatively complex specification for CnsM'.

Again, you should try to get the complete picture in your mind: As always, the key is that consistent sets have models. If $\Gamma \cup \{{\sim}\mathcal{P}\}$ is not consistent, then there is a derivation of $\mathcal{P}$ from Γ. So if there is no derivation of $\mathcal{P}$ from Γ, then $\Gamma \cup \{{\sim}\mathcal{P}\}$ is consistent, and so has a model — and the existence of a model for $\Gamma \cup \{{\sim}\mathcal{P}\}$ is sufficient to show that $\Gamma \nvDash \mathcal{P}$. Put the other way around, if $\Gamma \vDash \mathcal{P}$, then there is a derivation of $\mathcal{P}$ from Γ. We get the key point, that consistent sets have models, by finding a relation between consistent, and maximal consistent scapegoat sets. If a set is a maximal consistent scapegoat set, then it contains enough information to specify a model for the whole. The model for the big set then guarantees the existence of a model M for the original Γ.

E10.27. Return to the case from E10.20 on p. 497, but dropping the assumptions that there is no symbol for equality, and that $\mathcal{L}$ is identical to $\mathcal{L}'$. Add to the derivation system axioms,

A3 $\vdash t = t$

A4 $\vdash r = s \rightarrow (\mathcal{P} \rightarrow \mathcal{P}^{r}/_{s})$ — where s is free for replaced instances of r in $\mathcal{P}$

Provide a complete demonstration that this version of *A4* is adequate. You may appeal to any results from the text whose demonstration remains unchanged, but should recreate parts whose demonstration is not the same. Hint: You may find it helpful to demonstrate a relation to T8.5 as follows,

T8.5* For any formula $\mathcal{P}$, terms s and t, constant c, and variable x, $[\mathcal{P}^{s}/_{t}]^{c}_{x}$ is the same formula as $[\mathcal{P}^{c}_{x}]^{s^{c}_{x}}/_{t^{c}_{x}}$ — where the same instance(s) of s are replaced in each case.

E10.28. We have shown from T10.4 that if a set of formulas has a model, then it is consistent; and now that if an arbitrary set of formulas is consistent, then it has a model — and one whose U is this set of sets of positive integers. Notice that any such U is *countable* insofar as its members can be put into correspondence with the integers (we might, say, order the members by their least elements). Considering what we showed in the more on countability reference on p. 48, how might this be a problem for the logic of real numbers? Hint: Think about the consequences sentences in an arbitrary Γ may have about the number of elements in U.

E10.29. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The soundness of a derivation system, and its demonstration by mathematical induction.

b. The adequacy of a derivation system, and the basic strategy for its demonstration.

c. Maximality and consistency, and the reasons for them.

d. Scapegoat sets, and the reasons for them.

Theorems of Chapter 10

T10.1 For any interpretation I, variable assignment d, with terms t and r, if $\mathsf{I_d}[r] = \mathsf{o}$, then $\mathsf{I_{d(x|o)}}[t] = \mathsf{I_d}[t^x_r]$.

T10.2 For any interpretation I, variable assignment d, term r, and formula $\mathcal{Q}$, if $\mathsf{I_d}[r] = \mathsf{o}$, and r is free for x in $\mathcal{Q}$, then $\mathsf{I_d}[\mathcal{Q}^x_r] = \mathsf{S}$ iff $\mathsf{I_{d(x|o)}}[\mathcal{Q}] = \mathsf{S}$.

T10.3 If $\Gamma \vdash_{AD} \mathcal{P}$, then $\Gamma \vDash \mathcal{P}$. (*Soundness*)

T10.4 If there is an interpretation M such that $\mathsf{M}[\Gamma] = \mathsf{T}$ (a *model* for Γ), then Γ is consistent.

T10.5 If there is an interpretation M such that $\mathsf{M}[\Gamma \cup \{\sim\mathcal{A}\}] = \mathsf{T}$, then $\Gamma \nvdash \mathcal{A}$.

T10.6$_s$ For any set of formulas Σ and sentence $\mathcal{P}$, if $\Sigma \nvdash \sim\mathcal{P}$, then $\Sigma \cup \{\mathcal{P}\}$ is consistent.

T10.6 For any set of formulas Σ and sentence $\mathcal{P}$, if $\Sigma \nvdash \sim\mathcal{P}$, then $\Sigma \cup \{\mathcal{P}\}$ is consistent.

T10.7$_s$ There is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2 \ldots$ of all formulas in $\mathcal{L}_s$.

T10.7 There is an enumeration $\mathcal{Q}_1, \mathcal{Q}_2 \ldots$ of all the formulas, terms, and the like, in $\mathcal{L}'$.

T10.8$_s$ If Σ' is consistent, then Σ'' is maximal and consistent.

T10.8 If Σ' is consistent, then Σ'' is a maximal, consistent, scapegoat set.

T10.9$_s$ If Σ' is consistent, then for any sentence $\mathcal{B}$, of $\mathcal{L}_s$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

T10.9 If Σ' is consistent, then for any sentence $\mathcal{B}$ of $\mathcal{L}'$, $\mathsf{M}'[\mathcal{B}] = \mathsf{T}$ iff $\Sigma'' \vdash \mathcal{B}$.

T10.10$_s$ If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($*$)

T10.10 If Σ' is consistent, then $\mathsf{M}'[\Sigma'] = \mathsf{T}$. ($\star$)

T10.11$_s$ If $\Gamma \vDash_s \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. (*sentential adequacy*)

T10.11 If $\Gamma \vDash \mathcal{P}$, then $\Gamma \vdash \mathcal{P}$. (*quantificational adequacy*)

T10.12 If D is a derivation from Σ', and x is a variable that does not appear in D, then for any constant a, D^a_x is a derivation from $\Sigma'\,^a_x$.

T10.13 If Σ is consistent, then Σ' is consistent.

T10.14 For any variable assignment d, and for any term t in $\mathcal{L}$, $\mathsf{M_d}[t] = \mathsf{M'_d}[t]$.

T10.15 For any variable assignment d, and for any formula $\mathcal{P}$ in $\mathcal{L}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M'_d}[\mathcal{P}] = \mathsf{S}$.

T10.16 If $\mathsf{M}'[\Sigma'] = \mathsf{T}$, then $\mathsf{M}[\Sigma] = \mathsf{T}$.

T10.17a If Σ is consistent, then Σ has a model M. ($\mathcal{L}$ without equality)

T10.17 If Σ is consistent, then Σ has a model M. ($\mathcal{L}$ unconstrained) ($\star\star$)

Chapter 11

More Main Results

In this chapter, we take up results which deepen our understanding of the power and limits of logic. The first sections restrict discussion to *sentential* forms, for discussion of *expressive completenes* and *independence*. Then we turn to discussion of the conditions under which models are *isomorphic*, and transition to a discussion of submodels, and especially the Löwenheim-Skolem theoremes, which help us see some conditions under which models are not isomorphic.

> This chapter is not in finished form. It contains some parts which I've had occasion to write up — and found useful from time to time. But it's not worked into a fully-formed textbook chapter. Take it in the spirit with which it's provided!

11.1 Expressive Completeness

In chapter 5 on translation, we introduced the idea of a truth functional operator, where the truth value of the whole is a function of the truth values of the parts. We exhibited operators as truth functional by tables. Thus, if some ordinary expression $\mathscr{P}$ with components $\mathscr{A}$ and $\mathscr{B}$ has table,

(A)

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{P}$
T	T	***T***
T	F	***F***
F	T	***F***
F	F	***F***

then it is truth functional. And we translate by an equivalent formal operator: in this case $\mathscr{A} \wedge \mathscr{B}$ does fine. Of course, not every such table, or truth function, is directly represented by one of our operators. Thus, if $\mathscr{P}$ is 'neither $\mathscr{A}$ nor $\mathscr{B}$' we have the table,

(B)

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{P}$
T	T	***F***
T	F	***F***
F	T	***F***
F	F	***T***

where none of our operators is equivalent to this. But it takes only a little ingenuity to see that, say, $(\sim\mathscr{A} \wedge \sim\mathscr{B})$ or $\sim(\mathscr{A} \vee \mathscr{B})$ have the same table, and so result in a good translation. In chapter 5 (p. 158), we claimed that for any table a truth functional operator may have, there is always some way to generate that table by means of our formal operators — and, in fact, by means of just the operators $\sim$ and $\wedge$, or just the operators $\sim$ and $\vee$, or just the operators $\sim$ and $\rightarrow$. As it turns out, it is also possible to express any truth function by means of just the operator $\mid$. In this section, we prove these results. First,

T11.1. It is possible to represent any truth function by means of an expression with just the operators $\sim$, $\wedge$, and $\vee$.

The proof of this result is simple. Given an arbitrary truth function, we provide a recipe for constructing an expression with the same table. Insofar as for any truth function it is always possible to construct an expression with the same table, there must always be a formal expression with the same table.

Suppose we are given an arbitrary truth function, in this case with four basic sentences as on the left.

(C)

	$\mathscr{S}_1$	$\mathscr{S}_2$	$\mathscr{S}_3$	$\mathscr{S}_4$	$\mathscr{P}$	
1	T	T	T	T	***F***	$\mathscr{C}_1 = \mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \mathscr{S}_3 \wedge \mathscr{S}_4$
2	T	T	T	F	***F***	$\mathscr{C}_2 = \mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \mathscr{S}_3\sim \wedge \mathscr{S}_4$
3	T	T	F	T	***T***	$\mathscr{C}_3 = \mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \mathscr{S}_4$
4	T	T	F	F	***F***	$\mathscr{C}_4 = \mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \sim\mathscr{S}_4$
5	T	F	T	T	***T***	$\mathscr{C}_5 = \mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \mathscr{S}_3 \wedge \mathscr{S}_4$
6	T	F	T	F	***F***	$\mathscr{C}_6 = \mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \mathscr{S}_3 \wedge \sim\mathscr{S}_4$
7	T	F	F	T	***F***	$\mathscr{C}_7 = \mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \mathscr{S}_4$
8	T	F	F	F	***F***	$\mathscr{C}_8 = \mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \sim\mathscr{S}_4$
9	F	T	T	T	***F***	$\mathscr{C}_9 = \sim\mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \mathscr{S}_3 \wedge \mathscr{S}_4$
10	F	T	T	F	***F***	$\mathscr{C}_{10} = \sim\mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \mathscr{S}_3 \wedge \sim\mathscr{S}_4$
11	F	T	F	T	***F***	$\mathscr{C}_{11} = \sim\mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \mathscr{S}_4$
12	F	T	F	F	***T***	$\mathscr{C}_{12} = \sim\mathscr{S}_1 \wedge \mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \sim\mathscr{S}_4$
13	F	F	T	T	***T***	$\mathscr{C}_{13} = \sim\mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \mathscr{S}_3 \wedge \mathscr{S}_4$
14	F	F	T	F	***F***	$\mathscr{C}_{14} = \sim\mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \mathscr{S}_3 \wedge \sim\mathscr{S}_4$
15	F	F	F	T	***F***	$\mathscr{C}_{15} = \sim\mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \mathscr{S}_4$
16	F	F	F	F	***F***	$\mathscr{C}_{16} = \sim\mathscr{S}_1 \wedge \sim\mathscr{S}_2 \wedge \sim\mathscr{S}_3 \wedge \sim\mathscr{S}_4$

For this sentence $\mathscr{P}$ with basic sentences $\mathscr{S}_1 \ldots \mathscr{S}_n$, begin by constructing the *characteristic* sentence $\mathscr{C}_j$ corresponding to each row: If the interpretation I_j corresponding

to row j has $\mathsf{I}_j[\mathcal{S}_i] = \mathsf{T}$, then let $\mathcal{S}'_i = \mathcal{S}_i$. If $\mathsf{I}_j[\mathcal{S}_i] = \mathsf{F}$, let $\mathcal{S}'_i = \sim\mathcal{S}_i$. Then the characteristic sentence $\mathcal{C}_j$ corresponding to I_j is the conjunction of each $\mathcal{S}'_i$. So $\mathcal{C}_j = \mathcal{S}'_1 \wedge \ldots \wedge \mathcal{S}'_n$ (with appropriate parentheses). These sentences are exhibited above. The characteristic sentences are true *only* on their corresponding rows. Thus $\mathcal{C}_4$ above is true only when $\mathsf{I}[\mathcal{S}_1] = \mathsf{T}$, $\mathsf{I}[\mathcal{S}_2] = \mathsf{T}$, $\mathsf{I}[\mathcal{S}_3] = \mathsf{F}$, and $\mathsf{I}[\mathcal{S}_4] = \mathsf{F}$.

Then, given the characteristic sentences, if $\mathcal{P}$ is F on every row, $\mathcal{S}_1 \wedge \sim\mathcal{S}_1$ has the same table as $\mathcal{P}$. Otherwise, where $\mathcal{P}$ is T on rows $a, b \ldots d$, $\mathcal{C}_a \vee \mathcal{C}_b \vee \ldots \mathcal{C}_d$ (with appropriate parentheses) has the same table as $\mathcal{P}$. Thus, for example, $\mathcal{C}_3 \vee \mathcal{C}_5 \vee \mathcal{C}_{12} \vee \mathcal{C}_{13}$, that is,

$$(\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \mathcal{S}_4) \vee (\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4) \vee (\sim\mathcal{S}_1 \wedge \mathcal{S}_2 \wedge \sim\mathcal{S}_3 \wedge \sim\mathcal{S}_4) \vee (\sim\mathcal{S}_1 \wedge \sim\mathcal{S}_2 \wedge \mathcal{S}_3 \wedge \mathcal{S}_4)$$

has the same table as $\mathcal{P}$. Inserting parentheses, the resultant table is,

(D)

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$(\mathcal{C}_3$	$\vee$	$\mathcal{C}_5)$	$\vee$	$(\mathcal{C}_{12}$	$\vee$	$\mathcal{C}_{13})$	$\mathcal{P}$
1	T	T	T	T	F	F	F	***F***	F	F	F	***F***
2	T	T	T	F	F	F	F	***F***	F	F	F	***F***
3	T	T	F	T	T	T	F	***T***	F	F	F	***T***
4	T	T	F	F	F	F	F	***F***	F	F	F	***F***
5	T	F	T	T	F	T	T	***T***	F	F	F	***T***
6	T	F	T	F	F	F	F	***F***	F	F	F	***F***
7	T	F	F	T	F	F	F	***F***	F	F	F	***F***
8	T	F	F	F	F	F	F	***F***	F	F	F	***F***
9	F	T	T	T	F	F	F	***F***	F	F	F	***F***
10	F	T	T	F	F	F	F	***F***	F	F	F	***F***
11	F	T	F	T	F	F	F	***F***	F	F	F	***F***
12	F	T	F	F	F	F	F	***T***	T	T	F	***T***
13	F	F	T	T	F	F	F	***T***	F	T	T	***T***
14	F	F	T	F	F	F	F	***F***	F	F	F	***F***
15	F	F	F	T	F	F	F	***F***	F	F	F	***F***
16	F	F	F	F	F	F	F	***F***	F	F	F	***F***

And we have constructed an expression with the same table as $\mathcal{P}$. And similarly for any truth function with which we are confronted. So given any truth function, there is a formal expression with the same table.

In a by-now familiar pattern, the expressions produced by this method are not particularly elegant or efficient. Thus for the table,

(E)

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{P}$
T	T	***T***
T	F	***F***
F	T	***T***
F	F	***T***

by our method we get the expression $(\mathcal{A} \wedge \mathcal{B}) \vee (\sim\mathcal{A} \wedge \mathcal{B}) \vee (\sim\mathcal{A} \wedge \sim\mathcal{B})$. It has the right table. But, of course, $\mathcal{A} \to \mathcal{B}$ is much simpler! The point is not that the

resultant expressions are elegant or efficient, but that for any truth function, there *exists* a formal expression that works the same way.

We have shown that we can represent any truth function by an expression with operators $\sim$, $\wedge$, and $\vee$. But any such expression is an abbreviation of one whose only operators are $\sim$ and $\rightarrow$. So we can represent any truth function by an expression with just operators $\sim$ and $\rightarrow$. And we can argue for other cases. Thus, for example,

T11.2. It is possible to represent any truth function by means of an expression with just the operators $\sim$ and $\wedge$.

Again, the proof is simple. Given T11.1, if we can show that any $\mathcal{P}$ whose operators are $\sim$, $\wedge$ and $\vee$ corresponds to a $\mathcal{P}^*$ whose operators are just $\sim$ and $\wedge$, such that $\mathcal{P}$ and $\mathcal{P}^*$ have the same table — such that $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$ for any I — we will have shown that any truth function can be represented by an expression with just $\sim$ and $\wedge$. To see that this is so, where $\mathcal{P}$ is an atomic $\mathcal{S}$, set $\mathcal{P}^* = \mathcal{S}$; where $\mathcal{P}$ is $\sim\mathcal{A}$, set $\mathcal{P}^* = \sim\mathcal{A}^*$; where $\mathcal{P}$ is $\mathcal{A} \wedge \mathcal{B}$, set $\mathcal{P}^* = \mathcal{A}^* \wedge \mathcal{B}^*$; and where $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$, set $\mathcal{P}^* = \sim(\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*)$. Suppose the only operators in $\mathcal{P}$ are $\sim$, $\wedge$, and $\vee$, and consider an arbitrary interpretation I.

Basis: Where $\mathcal{P}$ is a sentence letter $\mathcal{S}$, then $\mathcal{P}^*$ is $\mathcal{S}$. So $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

Assp: For any i, $0 \leq i < k$, if $\mathcal{P}$ has i operator symbols, then $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

Show: If $\mathcal{P}$ has k operator symbols, then $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \wedge \mathcal{B}$, or $\mathcal{A} \vee \mathcal{B}$ where $\mathcal{A}$ and $\mathcal{B}$ have $< k$ operator symbols.

($\sim$) Suppose $\mathcal{P}$ is $\sim\mathcal{A}$; then $\mathcal{P}^*$ is $\sim\mathcal{A}^*$. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\sim\mathcal{A}] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\mathcal{A}] = \mathsf{F}$; by assumption iff $\mathsf{I}[\mathcal{A}^*] = \mathsf{F}$; by ST($\sim$), iff $\mathsf{I}[\sim\mathcal{A}^*] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

($\wedge$) Suppose $\mathcal{P}$ is $\mathcal{A} \wedge \mathcal{B}$; then $\mathcal{P}^*$ is $\mathcal{A}^* \wedge \mathcal{B}^*$. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{A} \wedge \mathcal{B}] = \mathsf{T}$; by ST$'$($\wedge$), iff $\mathsf{I}[\mathcal{A}] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B}] = \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}^*] = \mathsf{T}$ and $\mathsf{I}[\mathcal{B}^*] = \mathsf{T}$; by ST$'$($\wedge$), iff $\mathsf{I}[\mathcal{A}^* \wedge \mathcal{B}^*] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

($\vee$) Suppose $\mathcal{P}$ is $\mathcal{A} \vee \mathcal{B}$; then $\mathcal{P}^*$ is $\sim(\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*)$. $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}[\mathcal{A} \vee \mathcal{B}] = \mathsf{T}$; by ST$'$($\vee$), iff $\mathsf{I}[\mathcal{A}] = \mathsf{T}$ or $\mathsf{I}[\mathcal{B}] = \mathsf{T}$; by assumption iff $\mathsf{I}[\mathcal{A}^*] = \mathsf{T}$ or $\mathsf{I}[\mathcal{B}^*] = \mathsf{T}$; by ST($\sim$), iff $\mathsf{I}[\sim\mathcal{A}^*] = \mathsf{F}$ or $\mathsf{I}[\sim\mathcal{B}^*] = \mathsf{F}$; by ST$'$($\wedge$), iff $\mathsf{I}[\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*] = \mathsf{F}$; by ST($\sim$), iff $\mathsf{I}[\sim(\sim\mathcal{A}^* \wedge \sim\mathcal{B}^*)] = \mathsf{T}$; iff $\mathsf{I}[\mathcal{P}^*] = \mathsf{T}$.

If $\mathcal{P}$ has k operator symbols then $\mathsf{I}[\mathcal{P}] = \mathsf{I}[\mathcal{P}^*]$.

Indct: For any $\mathscr{P}$, $\mathsf{I}[\mathscr{P}] = \mathsf{I}[\mathscr{P}^*]$.

So if the operators in $\mathscr{P}$ are $\sim$, $\wedge$ and $\vee$, there is a $\mathscr{P}^*$ with just operators $\sim$ and $\wedge$ that has the same table. Perhaps this was obvious as soon as we saw that $\sim(\sim\mathscr{A} \wedge \sim\mathscr{B})$ has the same table as $\mathscr{A} \vee \mathscr{B}$. Since we can represent any truth function by an expression whose only operators are $\sim$, $\wedge$ and $\vee$, and we can represent any such $\mathscr{P}$ by a $\mathscr{P}^*$ whose only operators are $\sim$ and $\wedge$, we can represent any truth function by an expression with just operators $\sim$ and $\wedge$. And, by similar reasoning, we can represent any truth function by expressions whose only operators are $\sim$ and $\vee$, and by expressions whose only operator is $\mid$. This is left for homework.

In E8.10, we showed that if the operators in $\mathscr{P}$ are limited to $\rightarrow$, $\wedge$, $\vee$, and $\leftrightarrow$ then when the interpretation of every atomic is T, the interpretation of $\mathscr{P}$ is T. Perhaps this is obvious by consideration of the tables. It follows that not every truth function can be represented by expressions whose only operators are $\rightarrow$, $\wedge$, $\vee$, and $\leftrightarrow$; for there is no way to represent a function that is F on the top row, when all the atomics are T. Though it is much more difficult to establish, we showed in E8.27 that any expression whose only operators are $\sim$ and $\leftrightarrow$ (with at least four rows in its truth table) has an even number of Ts and Fs under its main operator. It follows that not every truth function can be represented by expressions whose only operators are $\sim$ and $\leftrightarrow$.

E11.1. Use the method of this section to find expressions with tables corresponding to $\mathscr{P}_1$, $\mathscr{P}_2$, and $\mathscr{P}_3$. Then show on a table that your expression for $\mathscr{P}_1$ in fact has the same truth function as $\mathscr{P}_1$.

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{C}$	$\mathscr{P}_1$	$\mathscr{P}_2$	$\mathscr{P}_3$
T	T	T	F	T	F
T	T	F	T	T	F
T	F	T	T	F	T
T	F	F	F	F	F
F	T	T	F	F	T
F	T	F	T	F	F
F	F	T	F	F	T
F	F	F	T	F	T

E11.2. (i) Show that we can represent any truth function by expressions whose only operators are $\sim$ and $\vee$. (ii) Show that we can represent any truth function by expressions whose only operator is $\mid$. Hint: Given what we have shown above, it is enough to show that you can represent expressions whose only operators are $\sim$ and $\rightarrow$, or $\sim$ and $\wedge$.

E11.3. Show that it is not possible to represent arbitrary truth functions by expressions whose only operator is $\sim$. Hint: it is easy to show by induction that any such expression has at least one T and one F under its main operator.

11.2 Independence

As we have seen, axiomatic systems are convenient insofar as their compact form makes reasoning about them relatively easy. Also, theoretically, axiomatic systems are attractive insofar as they expose what is at the base or foundation of logical systems. Given this latter aim, it is natural to wonder whether we could get the same results without one or more of our axioms. Say an axiom or rule is *independent* in a derivation system just in case its omission matters for what can be derived. In particular, then, an axiom is independent in a derivation system if *it* cannot be derived from the other axioms and rules. For suppose otherwise: that it can be derived from the other axioms and rules; then it is a theorem of the derivation system without the axiom, and any result of the system with the axiom can be derived using the theorem in place of the axiom; so the omission of the axiom does not matter for what can be derived, and the axiom is not independent. In this section, we show that A1, A2 and A3 of the sentential fragment of *AD* are independent of one another.

Say we want to show that A1 is independent of A2 and A3. When we showed, in chapter 8, that the sentential part of *AD* is weakly sound, we showed that A1, A2, A3 and their consequences have a certain feature — that there is no interpretation where a consequence is false. The basic idea here is to find a sort of "interpretation" on which A2, A3 and their consequences are sustained, but A1 is not. It follows that A1 is not among the consequences of A2 and A3, and so is independent of A2 and A3. Here is the key point: Any "interpretation" will do. In particular, consider the following tables which define a sort of numerical property for forms involving $\sim$ and $\to$.

A1($\sim$)

$\mathcal{P}$	$\sim\mathcal{P}$
0	1
1	1
2	0

A1($\to$)

$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{P} \to \mathcal{Q}$
0	0	0
0	1	2
0	2	2
1	0	2
1	1	2
1	2	0
2	0	0
2	1	0
2	2	0

Do not worry about what these tables "say"; it is sufficient that, given a numerical interpretation of the parts, we can always calculate the numerical value N of the whole. Thus, for example,

$$
\text{(F)}\qquad
\begin{tikzcd}[column sep=small]
A^{(0)} \arrow[d, dash] & & B^{(2)} \arrow[dl, dash] \\
{\sim}A^{(1)} \arrow[r, dash] & {\sim}A \to B^{(0)} &
\end{tikzcd}
$$

By A1($\sim$) row 1

By A1($\to$) row 6

if $\mathsf{N}[A] = 0$ and $\mathsf{N}[B] = 2$, then $\mathsf{N}[{\sim}A \to B] = 0$. The calculation is straightforward, based on the tables. And similarly for sentential forms of arbitrary complexity. Say a form is *select* iff it takes the value 0 on every numerical interpretation of its parts. (Compare the notion of semantic validity on which a form is valid iff it is T on every interpretation of its parts.) Again, do not worry about what the tables mean. They are constructed for the special purpose of demonstrating independence: We show that every consequence of A2 and A3 is select, but A1 is not. It follows that A1 is not a consequence of A2 and A3.

To see that A3 is select, and that A1 is not, all we have to do is complete the tables.

(G)

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \to$	$(\mathcal{B} \to \mathcal{A})$	$({\sim}\mathcal{B}$	$\to$	${\sim}\mathcal{A})$	$\to$	$[({\sim}\mathcal{B}$	$\to \mathcal{A})$	$\to \mathcal{B}]$
0	0	***0***	0	1	2	1	***0***	1	2	0
0	1	***2***	2	1	2	1	***0***	1	2	0
0	2	***0***	0	0	2	1	***0***	0	0	2
1	0	***0***	2	1	2	1	***0***	1	2	0
1	1	***0***	2	1	2	1	***0***	1	2	0
1	2	***2***	0	0	2	1	***0***	0	2	0
2	0	***0***	2	1	2	0	***0***	1	0	0
2	1	***0***	0	1	2	0	***0***	1	0	2
2	2	***0***	0	0	0	0	***0***	0	2	0

Since A1 has twos in the second and sixth rows, A1 is not select. Since A3 has zeros in every row, it is select. Alternatively, for A1, we might have reasoned as follows,

> Suppose $\mathsf{N}[\mathcal{A}] = 0$ and $\mathsf{N}[\mathcal{B}] = 1$. Then by A1($\to$), $\mathsf{N}[\mathcal{B} \to \mathcal{A}] = 2$; so by A1($\to$) again, $\mathsf{N}[\mathcal{A} \to (\mathcal{B} \to \mathcal{A})] = 2$. Since there is such an assignment, $\mathcal{A} \to (\mathcal{B} \to \mathcal{A})$ is not select.

And the result is the same. To see that A2 is select, again, it is enough to complete the table — it is painful, but we can do it:

(H)

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$(\mathcal{A} \rightarrow$	$(\mathcal{B} \rightarrow \mathcal{C}))$	$\rightarrow$	$((\mathcal{A} \rightarrow \mathcal{B})$	$\rightarrow$	$(\mathcal{A} \rightarrow \mathcal{C}))$
0	0	0	0	0	***0***	0	0	0
0	0	1	2	2	***0***	0	2	2
0	0	2	2	2	***0***	0	2	2
0	1	0	2	2	***0***	2	0	0
0	1	1	2	2	***0***	2	0	2
0	1	2	0	0	***0***	2	0	2
0	2	0	0	0	***0***	2	0	0
0	2	1	0	0	***0***	2	0	2
0	2	2	0	0	***0***	2	0	2
1	0	0	2	0	***0***	2	0	2
1	0	1	0	2	***0***	2	0	2
1	0	2	0	2	***0***	2	0	0
1	1	0	0	2	***0***	2	0	2
1	1	1	0	2	***0***	2	0	2
1	1	2	2	0	***0***	2	0	0
1	2	0	2	0	***0***	0	2	2
1	2	1	2	0	***0***	0	2	2
1	2	2	2	0	***0***	0	0	0
2	0	0	0	0	***0***	0	0	0
2	0	1	0	2	***0***	0	0	0
2	0	2	0	2	***0***	0	0	0
2	1	0	0	2	***0***	0	0	0
2	1	1	0	2	***0***	0	0	0
2	1	2	0	0	***0***	0	0	0
2	2	0	0	0	***0***	0	0	0
2	2	1	0	0	***0***	0	0	0
2	2	2	0	0	***0***	0	0	0

So both A2 and A3 are select. But now we are in a position to show,

T11.3. A1 is independent of A2 and A3.

Consider any derivation $\langle \mathcal{Q}_1, \mathcal{Q}_2 \ldots \mathcal{Q}_n \rangle$ where there are no premises, and the only axioms are instances of A2 and A3. By induction on line number, for any i, $\mathcal{Q}_i$ is select.

Basis: $\mathcal{Q}_1$ is an instance of A2 or A3, and as we have just seen, instances of A2 and A3 are select. So $\mathcal{Q}_1$ is select.

Assp: For any i, $0 \leq i < k$, $\mathcal{Q}_i$ is select.

Show: $\mathcal{Q}_k$ is select.

$\mathcal{Q}_k$ is an instance of A2 or A3 or arises from previous lines by MP. If $\mathcal{Q}_k$ is an instance of A2 or A3, then by reasoning as in the basis, $\mathcal{Q}_k$ is select. If $\mathcal{Q}_k$ arises from previous lines by MP, then the derivation has some lines,

$a.\ \mathscr{B}$
$b.\ \mathscr{B} \to \mathscr{C}$
$k.\ \mathscr{C}$ $\qquad a,b$ MP

where $a, b < k$ and $\mathscr{C}$ is $\mathscr{Q}_k$. By assumption, $\mathscr{B}$ and $\mathscr{B} \to \mathscr{C}$ are select. But by A1($\to$), both $\mathscr{B}$ and $\mathscr{B} \to \mathscr{C}$ evaluate to 0 only in the case when $\mathscr{C}$ also evaluates to 0; so if both $\mathscr{B}$ and $\mathscr{B} \to \mathscr{C}$ are select, then $\mathscr{C}$ is select as well. So $\mathscr{Q}_k$ is select.

Indct: For any n, $\mathscr{Q}_n$ is select.

So A1 cannot be derived from A2 and A3 — which is to say, A1 is independent of A2 and A3.

E11.4. Use the following tables to show that A2 is independent of A1 and A3.

A2($\sim$)

$\mathscr{P}$	$\sim\mathscr{P}$
0	1
1	0
2	1

A2($\to$)

$\mathscr{P}$	$\mathscr{Q}$	$\mathscr{P} \to \mathscr{Q}$
0	0	0
0	1	2
0	2	1
1	0	0
1	1	2
1	2	0
2	0	0
2	1	0
2	2	0

E11.5. Use the table method to show that A3 is independent of A1 and A2. That is, (i) find appropriate tables for $\sim$ and $\to$, and (ii) use your tables to show by induction that A3 is independent of A1 and A2. Hint: You do not need three-valued interpretations, and have already done the work in E8.13.

11.3 Isomorphic Models

Interpretations are *isomorphic* when they are structurally similar. Say a function f from r^{n} to s is *onto* set s just in case for each $\mathsf{o} \in \mathsf{s}$ there is some $\langle \mathsf{m}_1 \ldots \mathsf{m}_n \rangle \in \mathsf{r}^{\mathsf{n}}$ such that $\langle \langle \mathsf{m}_1 \ldots \mathsf{m}_n \rangle, \mathsf{o} \rangle \in \mathsf{f}$; a function is onto set s when it "reaches" every member of s. Then,

IS For some language $\mathcal{L}$, interpretation I is *isomorphic* to interpretation I' iff there is a 1:1 function ι (iota) from the universe of I onto the universe of I' where: for any sentence letter $\mathcal{S}$, $\mathsf{I}[\mathcal{S}] = \mathsf{I}'[\mathcal{S}]$; for any constant c, $\mathsf{I}[c] = \mathsf{m}$ iff $\mathsf{I}'[c] = \iota(\mathsf{m})$; for any relation symbol $\mathcal{R}^n$, $\langle \mathsf{m}_a \ldots \mathsf{m}_b\rangle \in \mathsf{I}[\mathcal{R}^n]$ iff $\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b)\rangle \in \mathsf{I}'(\mathcal{R}^n)$; and for any function symbol h^n, $\langle\langle \mathsf{m}_a \ldots \mathsf{m}_b\rangle, \mathsf{o}\rangle \in \mathsf{I}[h^n]$ iff $\langle\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b)\rangle, \iota(\mathsf{o})\rangle \in \mathsf{I}'[h^n]$.

If I is isomorphic to I', we write, $\mathsf{I} \cong \mathsf{I}'$. Notice that the condition on constants requires just that $\iota(\mathsf{I}[c]) = \mathsf{I}'[c]$; applying ι to the thing assigned to c by I, results in the thing assigned to c by I'. And similarly, the condition on function symbols requires that $\iota(\mathsf{I}[h^n]\langle \mathsf{m}_a \ldots \mathsf{m}_b\rangle) = \mathsf{I}'[h^n]\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b)\rangle$; for we have $\mathsf{I}[h^n]\langle \mathsf{m}_a \ldots \mathsf{m}_b\rangle = \mathsf{o}$, and $\iota(\mathsf{o}) = \mathsf{I}'[h^n]\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b)\rangle$. We might think of the two interpretations as already existing, and *finding* a function ι to exhibit them as isomorphic. Alternatively, given an interpretation I, and function ι from the universe of I onto some set U', we might think of I' as resulting from application of ι to I.

Here are some examples. In the first, it is perhaps particularly obvious that I and I' have the required structural similarity.

$$\text{(I)} \quad \begin{array}{lcccc} \mathsf{U}: & \text{Rover} & \text{Fido} & \text{Morris} & \text{Sylvester} \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathsf{U}': & \text{Ralph} & \text{Fredo} & \text{Manny} & \text{Salvador} \end{array}$$

$\mathsf{U} = \{$Rover, Fido, Morris, Sylvester$\}$. As represented by the arrows, function ι maps these onto a disjoint set U'. Then given I as below on the left, the corresponding isomorphic interpretation is I' as on the right.

$\mathsf{I}[r]$ = Rover	$\mathsf{I}'[r]$ = Ralph
$\mathsf{I}[m]$ = Morris	$\mathsf{I}'[m]$ = Manny
$\mathsf{I}[D]$ = {Rover, Fido}	$\mathsf{I}'[D]$ = {Ralph, Fredo}
$\mathsf{I}[C]$ = {Morris, Sylvester}	$\mathsf{I}'[C]$ = {Manny, Salvador}
$\mathsf{I}[P]$ = {⟨Rover, Morris⟩, ⟨Fido, Sylvester⟩}	$\mathsf{I}'[P]$ = {⟨Ralph, Manny⟩, ⟨Fredo, Salvador⟩}

On interpretation I, where Rover and Fido are dogs, and Morris and Sylvester are cats, we have that every dog pursues at least one cat. And, supposing that Ralph and Fredo are dogs, and Manny and Salvador are cats, the same properties and relations are preserved on I' — with only the particular individuals changed.

For a second case, let U be the same, but U' the very same set, only permuted or shuffled so that each object in U has a mate in U'.

$$\text{(J)} \quad \begin{array}{lcccc} \mathsf{U}: & \text{Rover} & \text{Fido} & \text{Morris} & \text{Sylvester} \\ & \downarrow & \downarrow & \downarrow & \downarrow \\ \mathsf{U}': & \text{Rover} & \text{Morris} & \text{Fido} & \text{Sylvester} \end{array}$$

So ι maps members of U to members of the very same set. Then given I as before, the corresponding isomorphic interpretation is I' is as follows.

$\mathsf{I}[r] = \text{Rover}$	$\mathsf{I}'[r] = \text{Rover}$
$\mathsf{I}[m] = \text{Morris}$	$\mathsf{I}'[m] = \text{Fido}$
$\mathsf{I}[D] = \{\text{Rover, Fido}\}$	$\mathsf{I}'[D] = \{\text{Rover, Morris}\}$
$\mathsf{I}[C] = \{\text{Morris, Sylvester}\}$	$\mathsf{I}'[C] = \{\text{Fido, Sylvester}\}$
$\mathsf{I}[P] = \{\langle\text{Rover, Morris}\rangle, \langle\text{Fido, Sylvester}\rangle\}$	$\mathsf{I}'[P] = \{\langle\text{Rover, Fido}\rangle, \langle\text{Morris, Sylvester}\rangle\}$

This time, there is no simple way to understand $\mathsf{I}'[D]$ as the set of all dogs, and $\mathsf{I}'[C]$ as the set of all cats. And we cannot say that the interpretation of P reflects dogs pursuing cats. But Morris *plays the same role* in I' as Fido in I; and similarly Fido plays the same role in I' as Morris in I. Thus, on I', each thing in the interpretation of D is such that it stands in the relation P to at least one thing in the interpretation of C — and this is just as in interpretation I.

A final example switches to $\mathscr{L}_{NT}^{<}$ and has an infinite U. We let U be the set $\mathbb{N}$ of natural numbers, U' be the set $\mathbb{P}$ of positive integers, and ι be the function $n + 1$.

$$\text{(K)} \quad \begin{array}{llllll} \mathsf{U}: & 0 & 1 & 2 & 3 & \dots \\ & \downarrow & \downarrow & \downarrow & \downarrow & \\ \mathsf{U}': & 1 & 2 & 3 & 4 & \dots \end{array}$$

Then where N is the standard interpretation for symbols of $\mathscr{L}_{NT}^{<}$,

$\mathsf{N}[\emptyset] = 0$

$\mathsf{N}[<] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ is less than } \mathsf{n}\}$

$\mathsf{N}[S] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}, \text{ and } \mathsf{n} \text{ is the successor of } \mathsf{m}\}$

$\mathsf{N}[+] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ plus } \mathsf{n} \text{ equals } \mathsf{o}\}$

we obtain N' as follows,

$\mathsf{N}'[\emptyset] = 1$

$\mathsf{N}'[<] = \{\langle \mathsf{m}+1, \mathsf{n}+1\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ is less than } \mathsf{n}\}$

$\mathsf{N}'[S] = \{\langle \mathsf{m}+1, \mathsf{n}+1\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}, \text{ and } \mathsf{n} \text{ is the successor of } \mathsf{m}\}$

$\mathsf{N}'[+] = \{\langle\langle \mathsf{m}+1, \mathsf{n}+1\rangle, \mathsf{o}+1\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}, \text{ and } \mathsf{m} \text{ plus } \mathsf{n} \text{ equals } \mathsf{o}\}$

Observe that anything in N' is taken from $\mathbb{P}$. In this case, we build N' explicitly by the rule for isomorphisms — simply finding $\iota(\mathsf{m}) = \mathsf{m} + 1$ from the corresponding element of N.

11.3.1 Isomorphism implies Equivalence

Given these examples, perhaps it is obvious that when interpretations are isomorphic, they make all the same formulas true.[1] Say,

EE For some language $\mathcal{L}$, interpretations I and I' are *elementarily equivalent* iff for any formula $\mathcal{P}$, $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{I}'[\mathcal{P}] = \mathsf{T}$.

If I is elementarily equivalent to I', write $\mathsf{I} \equiv \mathsf{I}'$. We show that isomorphic interpretations are elementarily equivalent. This is straightforward given a matched pair of results, of the sort we have often seen before.

T11.4. For some language $\mathcal{L}$, if interpretations $\mathsf{D} \cong \mathsf{H}$, and assignments d for D and h for H are such that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$, then for any term t, $\iota(\mathsf{D_d}[t]) = \mathsf{H_h}[t]$.

Suppose $\mathsf{D} \cong \mathsf{H}$, and corresponding assignments d and h are such that for any x, $\iota(\mathsf{d}(x)) = \mathsf{h}(x)$. By induction on the number of operator symbols in t.

Basis: If t has no function symbols, then it is a variable or a constant. If t is a variable x, then by TA(v), $\mathsf{D_d}[x] = \mathsf{d}(x)$; so $\iota(\mathsf{D_d}[x]) = \iota(\mathsf{d}[x])$; but we have supposed $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$; and by TA(v) again, $\mathsf{h}[x] = \mathsf{H_h}[x]$; so $\iota(\mathsf{D_d}[x]) = \mathsf{H_h}[x]$. If t is a constant c, then by TA(c), $\mathsf{D_d}[c] = \mathsf{D}[c]$; so $\iota(\mathsf{D_d}[c]) = \iota(\mathsf{D}[c])$; but since $\mathsf{D} \cong \mathsf{H}$, $\iota(\mathsf{D}[c]) = \mathsf{H}[c]$; and by TA(c) again, $\mathsf{H}[c] = \mathsf{H_h}[c]$; so $\iota(\mathsf{D_d}[c]) = \mathsf{H_h}[c]$.

Assp: For any i, $0 \le i < k$ if t has i function symbols, then $\iota(\mathsf{D_d}[t]) = \mathsf{H_h}[t]$.

Show: If t has k function symbols, then $\iota(\mathsf{D_d}[t]) = \mathsf{H_h}[t]$.

If t has k function symbols, then it is of the form $h^n t_1 \dots t_n$ for relation symbol h^n and terms $t_1 \dots t_n$ with $< k$ function symbols. Then $\mathsf{D_d}[t] = \mathsf{D_d}[h^n t_1 \dots t_n]$; by TA(f), $\mathsf{D_d}[h^n t_1 \dots t_n] = \mathsf{D}[h^n]\langle \mathsf{D_d}[\mathsf{t}_1] \dots \mathsf{D_d}[\mathsf{t}_n]\rangle$. So $\iota(\mathsf{D_d}[t]) = \iota(\mathsf{D}[h^n]\langle \mathsf{D_d}[\mathsf{t}_1] \dots \mathsf{D_d}[\mathsf{t}_n]\rangle)$; but since $\mathsf{D} \cong \mathsf{H}$, $\iota(\mathsf{D}[h^n]\langle \mathsf{D_d}[\mathsf{t}_1] \dots \mathsf{D_d}[\mathsf{t}_n]\rangle) = \mathsf{H}[h^n]\langle \iota(\mathsf{D_d}[\mathsf{t}_1]) \dots \iota(\mathsf{D_d}[\mathsf{t}_n])\rangle$; and by assumption, $\iota(\mathsf{D_d}[t_1]) = \mathsf{H_h}[t_1]$, and ... and $\iota(\mathsf{D_d}[t_n]) = \mathsf{H_h}[t_n]$; so $\mathsf{H}[h^n]\langle \iota(\mathsf{D_d}[\mathsf{t}_1]) \dots \iota(\mathsf{D_d}[\mathsf{t}_n])\rangle = \mathsf{H}[h^n]\langle \mathsf{H_h}[\mathsf{t}_1] \dots \mathsf{H_h}[\mathsf{t}_n]\rangle$; and by TA(f), $\mathsf{H}[h^n]\langle \mathsf{H_h}[\mathsf{t}_1] \dots \mathsf{H_h}[\mathsf{t}_n]\rangle = \mathsf{H_h}[h^n t_1 \dots t_n]$; which is just $\mathsf{H_h}[t]$; so $\iota(\mathsf{D_d}[t]) = \mathsf{H_h}[t]$.

[1] In *Reason, Truth and History*, Hilary Putnam makes this point to show that truth values of sentences are not sufficient to fix the interpretation of a language. As we shall see in this section, the technical point is clear enough. It is another matter whether it bears the philosophical weight he means for it to bear!

Indct: For any t, $\iota(\mathsf{D_d}[t]) = \mathsf{H_h}[t]$.

So when D and H are isomorphic, and for any variable x, ι maps $\mathsf{d}[x]$ to $\mathsf{h}[x]$, then for any term t, ι maps $\mathsf{D_d}[t]$ to $\mathsf{H_h}[t]$.

Now we are in a position to extend the result to one for satisfaction of formulas. If D and H are isomorphic, and for any variable x, ι maps $\mathsf{d}[x]$ to $\mathsf{h}[x]$, then a formula $\mathcal{P}$ will be satisfied on D with d just in case it is satisfied on H with h.

T11.5. For some language $\mathcal{L}$, if interpretations $\mathsf{D} \cong \mathsf{H}$, and assignments d for D and h for H are such that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$, then for any formula $\mathcal{P}$, $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$.

By induction on the number of operators in $\mathcal{P}$. Suppose $\mathsf{D} \cong \mathsf{H}$.

Basis: Suppose $\mathcal{P}$ has no operator symbols and d and h are such that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$. If $\mathcal{P}$ has no operator symbols, then it is sentence letter $\mathcal{S}$ or an atomic $\mathcal{R}^n t_1 \dots t_n$ for relation symbol $\mathcal{R}^n$ and terms $t_1 \dots t_n$. Suppose the former; then by SF(s), $\mathsf{D_d}[\mathcal{S}] = \mathsf{S}$ iff $\mathsf{D}[\mathcal{S}] = \mathsf{T}$; since $\mathsf{D} \cong \mathsf{H}$ iff $\mathsf{H}[\mathcal{S}] = \mathsf{T}$; by SF(s), iff $\mathsf{H_h}[\mathcal{S}] = \mathsf{S}$. Suppose the latter; by SF(r), $\mathsf{D_d}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$ iff $\langle \mathsf{D_d}[t_1] \dots \mathsf{D_d}[t_n] \rangle \in \mathsf{D}[\mathcal{R}^n]$; since $\mathsf{D} \cong \mathsf{H}$, iff $\langle \iota(\mathsf{D_d}[t_1]) \dots \iota(\mathsf{D_d}[t_n]) \rangle \in \mathsf{H}[\mathcal{R}^n]$; since $\mathsf{D} \cong \mathsf{H}$ and $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$, by T11.4, iff $\langle \mathsf{H_h}[t_1] \dots \mathsf{H_h}[t_n]) \rangle \in \mathsf{H}[\mathcal{R}^n]$; by SF(r), iff $\mathsf{H_h}[\mathcal{R}^n t_1 \dots t_n] = \mathsf{S}$.

Assp: For any i, $0 \leq i < k$, for d and h such that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$ and $\mathcal{P}$ with i operator symbols, $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$.

Show: For d and h such that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$ and $\mathcal{P}$ with k operator symbols, $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \to \mathcal{B}$, or $\forall x \mathcal{A}$ for variable x and formulas $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols. Suppose for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$.

($\sim$) Suppose $\mathcal{P}$ is of the form $\sim\mathcal{A}$. Then $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\sim\mathcal{A}] = \mathsf{S}$; by SF($\sim$), iff $\mathsf{D_d}[\mathcal{A}] \neq \mathsf{S}$; by assumption, iff $\mathsf{H_h}[\mathcal{A}] \neq \mathsf{S}$; by SF($\sim$), iff $\mathsf{H_h}[\sim\mathcal{A}] = \mathsf{S}$; iff $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$.

($\to$) Homework.

($\forall$) Suppose $\mathcal{P}$ is of the form $\forall x \mathcal{A}$. Then $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{D_d}[\forall x \mathcal{A}] = \mathsf{S}$; by SF($\forall$), iff for any $\mathsf{m} \in \mathsf{U_D}$, $\mathsf{D_{d(x|m)}}[\mathcal{A}] = \mathsf{S}$. Similarly, $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{H_h}[\forall x \mathcal{A}] = \mathsf{S}$; by SF($\forall$), iff for any $\mathsf{n} \in \mathsf{U_H}$, $\mathsf{H_{h(x|n)}}[\mathcal{A}] = \mathsf{S}$. (i)

Suppose $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$ but $\mathsf{D_d}[\mathcal{P}] \neq \mathsf{S}$; then any $\mathsf{n} \in \mathsf{U_H}$ is such that $\mathsf{H}_{\mathsf{h}(x|\mathsf{n})}[\mathcal{A}] = \mathsf{S}$, but there is some $\mathsf{m} \in \mathsf{U_D}$ such that $\mathsf{D}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{A}] \neq \mathsf{S}$. From the latter, insofar as $\mathsf{d}(x|\mathsf{m})$ and $\mathsf{h}(x|\iota(\mathsf{m}))$ have each member related by ι, the assumption applies and, $\mathsf{H}_{\mathsf{h}(x|\iota(\mathsf{m}))}[\mathcal{A}] \neq \mathsf{S}$; so there is an $\mathsf{n} \in \mathsf{U_H}$ such that $\mathsf{H}_{\mathsf{h}(x|\mathsf{n})}[\mathcal{A}] \neq \mathsf{S}$; this is impossible; reject the assumption: if $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$, then $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$. (ii) Similarly, [by homework] if $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$, then $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$. Hint: given $\mathsf{h}(x|\mathsf{n})$, there must be an m such that $\iota(\mathsf{m}) = \mathsf{n}$; then $\mathsf{d}(x|\mathsf{m})$ and $\mathsf{h}(x|\mathsf{n})$ are related so that the assumption applies.

For d and h such that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$ and $\mathcal{P}$ with k operator symbols, $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$.

Indct: For d and h such that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$, and any $\mathcal{P}$, $\mathsf{D_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{H_h}[\mathcal{P}] = \mathsf{S}$.

As often occurs, the most difficult case is for the quantifier. The key is that the assumption applies to $\mathsf{D_d}[\mathcal{P}]$ and $\mathsf{H_h}[\mathcal{P}]$ for *any* assignments d and h related so that for any x, $\iota(\mathsf{d}[x]) = \mathsf{h}[x]$. Supposing that d and h are so related, there is no reason to think that $\mathsf{d}(x|\mathsf{m})$ and h remain in that relation. The problem is solved with a corresponding modification to h: with $\mathsf{d}(x|\mathsf{m})$; we modify h so that the assignment to x simply is $\iota(\mathsf{m})$. Thus $\mathsf{d}(x|\mathsf{m})$ and $\mathsf{h}(x|\iota(\mathsf{m}))$ are related so that the assumption applies.

Now it is a simple matter to show that isomorphic models are elementarily equivalent.

T11.6. If $\mathsf{D} \cong \mathsf{H}$, then $\mathsf{D} \equiv \mathsf{H}$.

Suppose $\mathsf{D} \cong \mathsf{H}$. By TI, $\mathsf{D}[\mathcal{P}] \neq \mathsf{T}$ iff there is some assignment d such that $\mathsf{D_d}[\mathcal{P}] \neq \mathsf{S}$; since $\mathsf{D} \cong \mathsf{H}$, where d and h are related as in T11.5, iff $\mathsf{H_h}[\mathcal{P}] \neq \mathsf{S}$; by TI, iff $\mathsf{H}[\mathcal{P}] \neq \mathsf{T}$. So $\mathsf{D}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{H}[\mathcal{P}] = \mathsf{T}$; and $\mathsf{D} \equiv \mathsf{H}$.

Thus it is only the structures of interpretations up to isomorphism that matter for the truth values of formulas. And such structures are completely sufficient to determine truth values of formulas. It is another question whether truth values of formulas are sufficient to determine models, even up to isomorphism.

*E11.6. Complete the proof of T11.5. You should set up the complete induction, but may refer to the text, as the text refers to homework.

E11.7. (i) Explain what truth value the sentence $\forall x(Dx \rightarrow \exists y(Cy \wedge Pxy))$ has on interpretation I and then I' in example (I). Explain what truth values it has on I and then I' in example (J). (ii) Explain what truth value the sentence $S\emptyset + S\emptyset = SS\emptyset$ has on interpretations N and N' in example (K). Are these results as you expect? Explain.

11.3.2 When Equivalence implies Isomorphism

It turns out that when the universe of discourse is finite, elementary equivalence is sufficient to show isomorphism. Suppose $\mathsf{U_D}$ is finite and interpretations D and H are elementarily equivalent, so that every formula has the same truth value on the two interpretations. We find a sequence of formulas which contain sufficient information to show that D and H are isomorphic.

For some language $\mathscr{L}$, suppose $\mathsf{D} \equiv \mathsf{H}$ and $\mathsf{U_D} = \{\mathsf{m}_1, \mathsf{m}_2 \ldots \mathsf{m}_n\}$. For an enumeration x_1, $x_2 \ldots$ of the variables, consider some assignment d such that $\mathsf{d}[x_1] = \mathsf{m}_1$, $\mathsf{d}[x_2] = \mathsf{m}_2$, and ... and $\mathsf{d}[x_n] = \mathsf{m}_n$, and let $\mathscr{C}_0$ be the open formula,

$$[(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge \ldots \wedge x_1 \neq x_n) \wedge (x_2 \neq x_3 \wedge \ldots \wedge x_2 \neq x_n) \wedge (x_{n-1} \neq x_n)] \wedge \\ \forall v(v = x_1 \vee v = x_2 \vee \ldots \vee v = x_n)$$

with appropriate parentheses. You should see this expression on analogy with quantity expressions from chapter 5 on translation. Its existential closure, that is, $\exists x_1 \exists x_2 \ldots x_n \mathscr{C}_0$ is true just when there are exactly n things.

Now consider an enumeration, $\mathscr{A}_1$, $\mathscr{A}_2 \ldots$ of those atomic formulas in $\mathscr{L}$ whose only variables are $x_1 \ldots x_n$. And set $\mathscr{C}_i = \mathscr{C}_{i-1} \wedge \mathscr{A}_i$ if $\mathsf{D_d}[\mathscr{A}_i] = \mathsf{S}$, and otherwise, $\mathscr{C}_i = \mathscr{C}_{i-1} \wedge \sim\mathscr{A}_i$. It is easy to see that for any i, $\mathsf{D_d}[\mathscr{C}_i] = \mathsf{S}$. The argument is by induction on i.

T11.7. For any i, $\mathsf{D_d}[\mathscr{C}_i] = \mathsf{S}$.

Basis: For any a and b such that $1 \leq a, b \leq n$ and $a \neq b$, since x_a and x_b are assigned distinct members of $\mathsf{U_D}$, $\mathsf{D_d}[x_a = x_b] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{D_d}[x_a \neq x_b] = \mathsf{S}$; so by repeated applications of SF($\wedge$), $\mathsf{D_d}[(x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge \ldots \wedge x_1 \neq x_n) \wedge (x_2 \neq x_3 \wedge \ldots \wedge x_2 \neq x_n) \wedge (x_{n-1} \neq x_n)] = \mathsf{S}$. And since each member of $\mathsf{U_D}$ is assigned to some variable in $x_1 \ldots x_n$, for any $\mathsf{m} \in \mathsf{U_D}$, there is some a, $1 \leq a \leq n$ such that $\mathsf{D_{d(v|m)}}[v = x_a] = \mathsf{S}$. So by repeated applications of SF($\vee$), for any $\mathsf{m} \in \mathsf{U_D}$, $\mathsf{D_{d(v|m)}}[v = x_1 \vee v = x_2 \vee \ldots v = x_n] = \mathsf{S}$; so by SF($\forall$), $\mathsf{D_d}[\forall v(v = x_1 \vee v = x_2 \vee \ldots v = x_n)] = \mathsf{S}$; so by SF($\wedge$), $\mathsf{D_d}[\mathscr{C}_0] = \mathsf{S}$.

Assp: For any i, $0 \leq i < k$, $\mathsf{D_d}[\mathcal{C}_i] = \mathsf{S}$.

Show: $\mathsf{D_d}[\mathcal{C}_k] = \mathsf{S}$.

$\mathcal{C}_k$ is of the form $\mathcal{C}_{k-1} \wedge \mathcal{A}_k$ or $\mathcal{C}_{k-1} \wedge \sim\mathcal{A}_k$. In the first case, by assumption, $\mathsf{D_d}[\mathcal{C}_{k-1}] = \mathsf{S}$, and by construction, $\mathsf{D_d}[\mathcal{A}_k] = \mathsf{S}$; so by SF($\wedge$), $\mathsf{D_d}[\mathcal{C}_{k-1} \wedge \mathcal{A}_k] = \mathsf{S}$; which is to say, $\mathsf{D_d}[\mathcal{C}_k] = \mathsf{S}$. In the second case, again $\mathsf{D_d}[\mathcal{C}_{k-1}] = \mathsf{S}$; and by construction, $\mathsf{D_d}[\mathcal{A}_k] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{D_d}[\sim\mathcal{A}_k] = \mathsf{S}$; so by SF($\wedge$), $\mathsf{D_d}[\mathcal{C}_{k-1} \wedge \sim\mathcal{A}_k] = \mathsf{S}$; which is to say, $\mathsf{D_d}[\mathcal{C}_k] = \mathsf{S}$.

Indct: For any i, $\mathsf{D_d}[\mathcal{C}_i] = \mathsf{S}$.

So these formulas, though increasingly long, are all satisfied on assignment d.

Now, for the specification of an isomorphism between the interpretations, we set out to show there is a corresponding assignment h on which all the same expressions are satisfied. First, for any $\mathcal{C}_i$, consider its existential closure, $\exists x_1 \ldots \exists x_n \mathcal{C}_i$. It is easy to see that for any $\mathcal{C}_i$, $\mathsf{H}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] = \mathsf{T}$. Suppose otherwise; then since $\mathsf{D} \equiv \mathsf{H}$, $\mathsf{D}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] \neq \mathsf{T}$; so by TI, there is some assignment d' such that $\mathsf{D_{d'}}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] \neq \mathsf{S}$; so, since the closure of $\mathcal{C}_i$ has no free variables, by T8.4, $\mathsf{D_d}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] \neq \mathsf{S}$; so by repeated application of SF($\exists$), $\mathsf{D_d}[\mathcal{C}_i] \neq \mathsf{S}$; but by T11.7, this is impossible; reject the assumption: $\mathsf{H}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] = \mathsf{T}$. When the existential is not satisfied on d, as we remove the quantifiers, in each case, the resultant formula without a quantifier is unsatisfied on $\mathsf{d}(x|\mathsf{m})$ for any $\mathsf{m} \in \mathsf{U_D}$; so it is unsatisfied when $\mathsf{m} = \mathsf{d}[x]$ — so that the formula without the quantifier is unsatisfied on the original d. Observe that there are thus exactly n members of $\mathsf{U_H}$: $\mathsf{H}[\exists x_1 \ldots \exists x_n \mathcal{C}_0] = \mathsf{T}$; and, as we have already noted, this can be the case iff there are exactly n members of $\mathsf{U_H}$.

Now for some assignment h', let h range over assignments that differ from h' at most in assignment to $x_1 \ldots x_n$. Set $\Omega_i = \{\mathsf{h} \mid \mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}\}$, and $\Omega = \bigcap_{i \geq 0} \Omega_i$. Observe: (i) No Ω_i is empty. Since $\mathsf{H}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] = \mathsf{T}$, by TI, for any assignment h^*, $\mathsf{H_{h^*}}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] = \mathsf{S}$; so $\mathsf{H_{h'}}[\exists x_1 \ldots \exists x_n \mathcal{C}_i] = \mathsf{S}$; so by repeated applications of SF($\exists$), there is some h such that $\mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}$. When the quantifiers come off, the result is some assignment that differs at most in assignments to $x_1 \ldots x_n$ and so some assignment in Ω_i. (ii) For any $j \geq i$, $\Omega_j \subseteq \Omega_i$. Suppose otherwise; then there is some h such that $\mathsf{h} \in \Omega_j$ but $\mathsf{h} \notin \Omega_i$; so by construction, $\mathsf{H_h}[\mathcal{C}_j] = \mathsf{S}$ but $\mathsf{H_h}[\mathcal{C}_i] \neq \mathsf{S}$; if $j = i$ this is impossible; so suppose $j > i$; then $\mathcal{C}_j$ is of the sort, $\mathcal{C}_i \wedge \mathcal{B}_{i+1} \wedge \mathcal{B}_{i+2} \wedge \ldots \wedge \mathcal{B}_j$ where $\mathcal{B}_{i+1} \ldots \mathcal{B}_j$ are either atomics or negated atomics; so by repeated application of SF($\wedge$), $\mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}$; this is impossible; reject the assumption: $\Omega_j \subseteq \Omega_i$. (iii) Finally, there are at most finitely many assignments

of the sort h. Since any h differs from h' at most in assignments to $x_1 \dots x_n$, and there are just n members of $\mathsf{U_H}$, there are n^n assignments of the sort h.

From these results it follows that Ω is non-empty. Suppose otherwise. Then for any h, there is some Ω_i such that $\mathsf{h} \notin \Omega_i$. But there are only finitely many assignments of the sort h. So we may consider finitely many $\Omega_a \dots \Omega_b$ from which for any h there is some Ω_i such that $\mathsf{h} \notin \Omega_i$. But where each subscript in $a \dots b$ is $\leq b$, for each Ω_i, $\Omega_b \subseteq \Omega_i$; and since each h is missing from at least one Ω_i, we have that Ω_b is therefore empty. Ω_b must lack each of the assignments missing from prior members of the sequence. But this is impossible; reject the assumption: Ω is not empty. So we have what we wanted: any h in Ω is an assignment that satisfies every $\mathcal{C}_i$.

Now we are ready to specify a mapping for our isomorphism! Indeed, we are ready to show,

T11.8. If $\mathsf{D} \equiv \mathsf{H}$ and $\mathsf{U_D}$ is finite, then $\mathsf{D} \cong \mathsf{H}$.

Suppose $\mathsf{D} \equiv \mathsf{H}$ and $\mathsf{U_D}$ is finite. Then there are Ω and formulas $\mathcal{C}_i$ as above. For some particular $\mathsf{h} \in \Omega$, for any i, $1 \leq i \leq n$, let $\iota(\mathsf{d}[x_i]) = \mathsf{h}[x_i]$. Since $\mathsf{h} \in \Omega$, for any $\mathcal{C}_i$, $\mathsf{H_h}[\mathcal{C}_i] = \mathsf{S}$. So $\mathsf{H_h}[\mathcal{C}_0] = \mathsf{S}$. So h assigns each x_i to a different member of $\mathsf{U_H}$, and ι is onto $\mathsf{U_H}$, as it should be. We now set out to show that the other conditions for isomorphism are met.

Sentence letters. Since $\mathsf{D} \equiv \mathsf{H}$, for any sentence letter $\mathcal{S}$, $\mathsf{D}[\mathcal{S}] = \mathsf{T}$; iff $\mathsf{H}[\mathcal{S}] = \mathsf{T}$; so $\mathsf{D}[\mathcal{S}] = \mathsf{H}[\mathcal{S}]$.

Constants. We require that for any constant c, $\mathsf{D}[c] = \mathsf{m}_i$ iff $\mathsf{H}[c] = \iota(\mathsf{m}_i)$. (i) For some constant c, suppose $\mathsf{D}[c] = \mathsf{m}_i$. Since $\mathsf{d}[x_i] = \mathsf{m}_i$, $\iota(\mathsf{m}_i) = \iota(\mathsf{d}[x_i]) = \mathsf{h}[x_i]$. By TA(c), $\mathsf{D_d}[c] = \mathsf{D}[c] = \mathsf{m}_i$; and by TA(v), $\mathsf{D_d}[x_i] = \mathsf{d}[x_i] = \mathsf{m}_i$; so $\mathsf{D_d}[c] = \mathsf{D_d}[x_i]$; so $\langle \mathsf{D_d}[c], \mathsf{D_d}[x_i] \rangle \in \mathsf{D}[=]$; so by SF(r), $\mathsf{D_d}[c = x_i] = \mathsf{S}$; so $c = x_i$ is a conjunct in some $\mathcal{C}_n$; but $\mathsf{H_h}[\mathcal{C}_n] = \mathsf{S}$; so by repeated applications of SF($\wedge$), $\mathsf{H_h}[c = x_i] = \mathsf{S}$; so by SF(r), $\langle \mathsf{H_h}[c], \mathsf{H_h}[x_i] \rangle \in \mathsf{H}[=]$; so $\mathsf{H_h}[c] = \mathsf{H_h}[x_i]$; but by TA(c), $\mathsf{H_h}[c] = \mathsf{H}[c]$, and by TA(v), $\mathsf{H_h}[x_i] = \mathsf{h}[x_i]$; so $\mathsf{H}[c] = \mathsf{h}[x_i]$; so $\mathsf{H}[c] = \iota(\mathsf{m}_i)$.

(ii) Suppose $\mathsf{D}[c] \neq \mathsf{m}_i$. As before, $\iota(\mathsf{m}_i) = \mathsf{h}[x_i]$; and $\mathsf{D_d}[x_i] = \mathsf{m}_i$. But by TA(c), $\mathsf{D_d}[c] = \mathsf{D}[c]$; so $\mathsf{D_d}[c] \neq \mathsf{m}_i$; so $\mathsf{D_d}[c] \neq \mathsf{D_d}[x_i]$; so $\langle \mathsf{D_d}[c], \mathsf{D_d}[x_i] \rangle \notin \mathsf{D}[=]$; so by SF(r), $\mathsf{D_d}[c = x_i] \neq \mathsf{S}$; so $c \neq x_i$ is a conjunct in some $\mathcal{C}_n$; but $\mathsf{H_h}[\mathcal{C}_n] = \mathsf{S}$; so by repeated applications of SF($\wedge$), $\mathsf{H_h}[c \neq x_i] = \mathsf{S}$; so by SF($\sim$), and SF(r), $\langle \mathsf{H_h}[c], \mathsf{H_h}[x_i] \rangle \notin \mathsf{H}[=]$; so $\mathsf{H_h}[c] \neq \mathsf{H_h}[x_i]$; but by TA(c), $\mathsf{H_h}[c] = \mathsf{H}[c]$, and by TA(v), $\mathsf{H_h}[x_i] = \mathsf{h}[x_i]$; so $\mathsf{H}[c] \neq \mathsf{h}[x_i]$; so $\mathsf{H}[c] \neq \iota(\mathsf{m}_i)$.

Relation Symbols. We require that for any relation symbol $\mathcal{R}^n$, $\langle \mathsf{m}_a \ldots \mathsf{m}_b \rangle \in \mathsf{D}[\mathcal{R}^n]$ iff $\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b) \rangle \in \mathsf{H}(\mathcal{R}^n)$. (i) Suppose $\langle \mathsf{m}_a \ldots \mathsf{m}_b \rangle \in \mathsf{D}[\mathcal{R}^n]$. Since $\mathsf{d}[x_a] = \mathsf{m}_a$, and …and $\mathsf{d}[x_b] = \mathsf{m}_b$ we have, $\iota(\mathsf{m}_a) = \iota(\mathsf{d}[x_a]) = \mathsf{h}[x_a]$, and …and $\iota(\mathsf{m}_b) = \iota(\mathsf{d}[x_b]) = \mathsf{h}[x_b]$, and also by TA(v), $\mathsf{D}_\mathsf{d}[x_a] = \mathsf{m}_a$, and …and $\mathsf{D}_\mathsf{d}[x_b] = \mathsf{m}_b$; so $\langle \mathsf{D}_\mathsf{d}[x_a], \ldots \mathsf{D}_\mathsf{d}[x_b] \rangle \in \mathsf{D}[\mathcal{R}^n]$; so by SF(r), $\mathsf{D}_\mathsf{d}[\mathcal{R}^n x_a \ldots x_b] = \mathsf{S}$; so $\mathcal{R}^n x_a \ldots x_b$ is a conjunct of some $\mathcal{C}_n$; but $\mathsf{H}_\mathsf{h}[\mathcal{C}_n] = \mathsf{S}$; so by repeated applications of SF($\wedge$), $\mathsf{H}_\mathsf{h}[\mathcal{R}^n x_a \ldots x_b] = \mathsf{S}$; so by SF(r), $\langle \mathsf{H}_\mathsf{h}[x_a], \ldots \mathsf{H}_\mathsf{h}[x_b] \rangle \in \mathsf{H}[\mathcal{R}^n]$; but by TA(v), $\mathsf{H}_\mathsf{h}[x_a] = \mathsf{h}[x_a] = \iota(\mathsf{m}_a)$, and …and $\mathsf{H}_\mathsf{h}[x_b] = \mathsf{h}[x_b] = \iota(\mathsf{m}_b)$; so $\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b) \rangle \in \mathsf{H}[\mathcal{R}^n]$.

(ii) Suppose $\langle \mathsf{m}_a \ldots \mathsf{m}_b \rangle \notin \mathsf{D}[\mathcal{R}^n]$. As before, $\iota(\mathsf{m}_a) = \mathsf{h}[x_a]$, and …and $\iota(\mathsf{m}_b) = \mathsf{h}[x_b]$; similarly, $\mathsf{D}_\mathsf{d}[x_a] = \mathsf{m}_a$, and …and $\mathsf{D}_\mathsf{d}[x_b] = \mathsf{m}_b$; so $\langle \mathsf{D}_\mathsf{d}[x_a], \ldots \mathsf{D}_\mathsf{d}[x_b] \rangle \notin \mathsf{D}[\mathcal{R}^n]$; so by SF(r), $\mathsf{D}_\mathsf{d}[\mathcal{R}^n x_a \ldots x_b] \neq \mathsf{S}$; and $\sim\mathcal{R}^n x_a \ldots x_b$ is a conjunct of some $\mathcal{C}_n$; but $\mathsf{H}_\mathsf{h}[\mathcal{C}_n] = \mathsf{S}$; so by repeated applications of SF($\wedge$), $\mathsf{H}_\mathsf{h}[\sim\mathcal{R}^n x_a \ldots x_b] = \mathsf{S}$; so by SF($\sim$) and SF(r), $\langle \mathsf{H}_\mathsf{h}[x_a], \ldots \mathsf{H}_\mathsf{h}[x_b] \rangle \notin \mathsf{H}[\mathcal{R}^n]$; but as before, $\mathsf{H}_\mathsf{h}[x_a] = \iota(\mathsf{m}_a)$, and …and $\mathsf{H}_\mathsf{h}[x_b] = \iota(\mathsf{m}_b)$; so $\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b) \rangle \notin \mathsf{H}[\mathcal{R}^n]$.

Function symbols. We require that for any function symbol $\hbar^n$, $\langle \langle \mathsf{m}_a \ldots \mathsf{m}_b \rangle, \mathsf{m}_c \rangle \in \mathsf{D}[\hbar^n]$ iff $\langle \langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b) \rangle, \iota(\mathsf{m}_c) \rangle \in \mathsf{H}[\hbar^n]$. (i) Suppose $\langle \langle \mathsf{m}_a \ldots \mathsf{m}_b \rangle, \mathsf{m}_c \rangle \in \mathsf{D}[\hbar^n]$. Since $\mathsf{d}[x_a] = \mathsf{m}_a$, and …and $\mathsf{d}[x_b] = \mathsf{m}_b$, and $\mathsf{d}[x_c] = \mathsf{m}_c$, we have, $\iota(\mathsf{m}_a) = \iota(\mathsf{d}[x_a]) = \mathsf{h}[x_a]$, and …and $\iota(\mathsf{m}_b) = \iota(\mathsf{d}[x_b]) = \mathsf{h}[x_b]$, and $\iota(\mathsf{m}_c) = \iota(\mathsf{d}[x_c]) = \mathsf{h}[x_c]$; and also by TA(v), $\mathsf{D}_\mathsf{d}[x_a] = \mathsf{m}_a$, and …and $\mathsf{D}_\mathsf{d}[x_b] = \mathsf{m}_b$, and $\mathsf{D}_\mathsf{d}[x_c] = \mathsf{m}_c$; so $\langle \langle \mathsf{D}_\mathsf{d}[x_a] \ldots \mathsf{D}_\mathsf{d}[x_b] \rangle, \mathsf{D}_\mathsf{d}[x_c] \rangle \in \mathsf{D}[\hbar^n]$; so $\mathsf{D}[\hbar^n]\langle \mathsf{D}_\mathsf{d}[x_a] \ldots \mathsf{D}_\mathsf{d}[x_b] \rangle = \mathsf{D}_\mathsf{d}[x_c]$; so by TA(f), $\mathsf{D}_\mathsf{d}[\hbar^n x_a \ldots x_b] = \mathsf{D}_\mathsf{d}[x_c]$; so $\langle \mathsf{D}_\mathsf{d}[\hbar^n x_a \ldots x_b], \mathsf{D}_\mathsf{d}[x_c] \rangle \in \mathsf{D}[=]$; so by SF(r), $\mathsf{D}_\mathsf{d}[\hbar^n x_a \ldots x_b = x_c] = \mathsf{S}$; so $\hbar^n x_a \ldots x_b = x_c$ is a conjunct of some $\mathcal{C}_n$; but $\mathsf{H}_\mathsf{h}[\mathcal{C}_n] = \mathsf{S}$; so by repeated applications of SF($\wedge$), $\mathsf{H}_\mathsf{h}[\hbar^n x_a \ldots x_b = x_c] = \mathsf{S}$; so by SF(r), $\langle \mathsf{H}_\mathsf{h}[\hbar^n x_a \ldots x_b], \mathsf{H}_\mathsf{h}[x_c] \rangle \in \mathsf{H}[=]$; so $\mathsf{H}_\mathsf{h}[\hbar^n x_a \ldots x_b] = \mathsf{H}_\mathsf{h}[x_c]$; but by TA(f), $\mathsf{H}_\mathsf{h}[\hbar^n x_a \ldots x_b] = \mathsf{H}[\hbar^n]\langle \mathsf{H}_\mathsf{h}[x_a] \ldots \mathsf{H}_\mathsf{h}[x_b] \rangle$; so $\mathsf{H}[\hbar^n]\langle \mathsf{H}_\mathsf{h}[x_a] \ldots \mathsf{H}_\mathsf{h}[x_b] \rangle = \mathsf{H}_\mathsf{h}[x_c]$; so $\langle \langle \mathsf{H}_\mathsf{h}[x_a] \ldots \mathsf{H}_\mathsf{h}[x_b] \rangle, \mathsf{H}_\mathsf{h}[x_c] \rangle \in \mathsf{H}[\hbar^n]$; but by TA(v), $\mathsf{H}_\mathsf{h}[x_a] = \mathsf{h}[x_a] = \iota(\mathsf{m}_a)$, and …$\mathsf{H}_\mathsf{h}[x_b] = \mathsf{h}[x_b] = \iota(\mathsf{m}_b)$, and $\mathsf{H}_\mathsf{h}[x_c] = \mathsf{h}[x_c] = \iota(\mathsf{m}_c)$; so $\langle \langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b) \rangle, \iota(\mathsf{m}_c) \rangle \in \mathsf{H}[\hbar^n]$.

(ii) Suppose $\langle \langle \mathsf{m}_a \ldots \mathsf{m}_b \rangle, \mathsf{m}_c \rangle \notin \mathsf{D}[\hbar^n]$. As before, $\iota(\mathsf{m}_a) = \mathsf{h}[x_a]$, and …and $\iota(\mathsf{m}_b) = \mathsf{h}[x_b]$, and $\iota(\mathsf{m}_c) = \mathsf{h}[x_c]$; and also $\mathsf{D}_\mathsf{d}[x_a] = \mathsf{m}_a$, and …and $\mathsf{D}_\mathsf{d}[x_b] = \mathsf{m}_b$, and $\mathsf{D}_\mathsf{d}[x_c] = \mathsf{m}_c$; so $\langle \langle \mathsf{D}_\mathsf{d}[x_a] \ldots \mathsf{D}_\mathsf{d}[x_b] \rangle, \mathsf{D}_\mathsf{d}[x_c] \rangle \notin \mathsf{D}[\hbar^n]$; so $\mathsf{D}[\hbar^n]\langle \mathsf{D}_\mathsf{d}[x_a] \ldots \mathsf{D}_\mathsf{d}[x_b] \rangle \neq \mathsf{D}_\mathsf{d}[x_c]$; so by TA(f), $\mathsf{D}_\mathsf{d}[\hbar^n x_a \ldots x_b] \neq \mathsf{D}_\mathsf{d}[x_c]$; so $\langle \mathsf{D}_\mathsf{d}[\hbar^n x_a \ldots x_b], \mathsf{D}_\mathsf{d}[x_c] \rangle \notin \mathsf{D}[=]$; so by SF(r), $\mathsf{D}_\mathsf{d}[\hbar^n x_a \ldots x_b = x_c] \neq \mathsf{S}$; so $\hbar^n x_a \ldots x_b \neq x_c$ is a conjunct of some $\mathcal{C}_n$; but $\mathsf{H}_\mathsf{h}[\mathcal{C}_n] = \mathsf{S}$;

so by repeated applications of SF($\wedge$), $\mathsf{H_h}[\hslash^n x_a \ldots x_b \neq x_c] = \mathsf{S}$; so by SF($\sim$) and SF(r), $\langle \mathsf{H_h}[\hslash^n x_a \ldots x_b], \mathsf{H_h}[x_c]\rangle \notin \mathsf{H}[=]$; so $\mathsf{H_h}[\hslash^n x_a \ldots x_b] \neq \mathsf{H_h}[x_c]$; but by TA(f), $\mathsf{H_h}[\hslash^n x_a \ldots x_b] = \mathsf{H}[\hslash^n]\langle \mathsf{H_h}[x_a] \ldots \mathsf{H_h}[x_b]\rangle$; and $\mathsf{H}[\hslash^n]\langle \mathsf{H_h}[x_a] \ldots \mathsf{H_h}[x_b]\rangle \neq \mathsf{H_h}[x_c]$; so $\langle\langle \mathsf{H_h}[x_a] \ldots \mathsf{H_h}[x_b]\rangle, \mathsf{H_h}[x_c]\rangle \notin \mathsf{H}[\hslash^n]$; but as before, $\mathsf{H_h}[x_a] = \iota(\mathsf{m}_a)$, and $\ldots \mathsf{H_h}[x_b] = \iota(\mathsf{m}_b)$, and $\mathsf{H_h}[x_c] = \iota(\mathsf{m}_c)$; so $\langle\langle \iota(\mathsf{m}_a) \ldots \iota(\mathsf{m}_b)\rangle, \iota(\mathsf{m}_c)\rangle \notin \mathsf{H}[\hslash^n]$.

Thus elementary equivalence is sufficient for isomorphism in the case where the universe of discourse is finite. This is an interesting result! Consider any interpretation D with a finite $\mathsf{U_D}$, and the set of formulas Δ (Delta) true on D. By our result, any other model H that makes all the formulas in Δ true — any H such that $\mathsf{D} \equiv \mathsf{H}$ — is such that D is isomorphic to H. As we shall shortly see, the situation is not so straightforward when $\mathsf{U_D}$ is infinite.

11.4 Compactness and Isomorphism

Compactness takes the link between syntax and semantics from adequacy, and combines it with the finite length of derivations. The result is simple enough, and puts us in a position to obtain a range of further conclusions.

ST A set Σ of formulas is *satisfiable* iff it has a model. Σ is *finitely satisfiable* iff every finite subset of it has a model.

Now compactness draws a connection between satisfiability, and finite satisfiability,

T11.9. A set of formulas Σ is satisfiable iff it is finitely satisfiable. (*compactness*)

(i) Suppose Σ is satisfiable, but not finitely satisfiable. Then there is some M such that $\mathsf{M}[\Sigma] = \mathsf{T}$; but there is a finite $\Sigma' \subseteq \Sigma$ such that any M' has $\mathsf{M}'[\Sigma'] \neq \mathsf{T}$; so $\mathsf{M}[\Sigma'] \neq \mathsf{T}$; so there is a formula $\mathcal{P} \in \Sigma'$ such that $\mathsf{M}[\mathcal{P}] \neq \mathsf{T}$; but since $\Sigma' \subseteq \Sigma$, $\mathcal{P} \in \Sigma$; so $\mathsf{M}[\Sigma] \neq \mathsf{T}$. This is impossible; reject the assumption: if Σ is satisfiable, then it is finitely satisfiable.

(ii) Suppose Σ is finitely satisfiable, but not satisfiable. By T10.17, if Σ is consistent, then it has a model M. But since Σ is not satisfiable, it has no model; so it is not consistent; so there is some formula $\mathcal{A}$ such that $\Sigma \vdash \mathcal{A}$ and $\Sigma \vdash \sim\mathcal{A}$; consider derivations of these results, and the set Σ^* of premises of these derivations; since derivations are finite, Σ^* is finite; and since Σ^* includes all the premises, $\Sigma^* \vdash \mathcal{A}$ and $\Sigma^* \vdash \sim\mathcal{A}$; so by soundness, $\Sigma^* \vDash \mathcal{A}$ and $\Sigma^* \vDash \sim\mathcal{A}$; since Σ is finitely satisfiable, there must be some model M^*

such that $\mathsf{M}^*[\Sigma^*] = \mathsf{T}$; then by QV, $\mathsf{M}^*[\mathcal{A}] = \mathsf{T}$ and $\mathsf{M}^*[\sim\mathcal{A}] = \mathsf{T}$. But by T7.5, there is no M^* and $\mathcal{A}$ such that $\mathsf{M}^*[\mathcal{A}] = \mathsf{T}$ and $\mathsf{M}^*[\sim\mathcal{A}] = \mathsf{T}$. This is impossible; reject the assumption: if Σ is finitely satisfiable, then it is satisfiable.

This theorem puts us in a position to reason from finite satisfiability to satisfiability. And the results of such reasoning may be startling. Consider again the standard interpretation $\mathsf{N1}$ for $\mathcal{L}_{NT}^{<}$,

$\mathsf{N}[\emptyset] = 0$

$\mathsf{N}[<] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and m is less than $\mathsf{n}\}$

$\mathsf{N}[S] = \{\langle \mathsf{m}, \mathsf{n}\rangle \mid \mathsf{m}, \mathsf{n} \in \mathbb{N}$, and n is the successor of $\mathsf{m}\}$

$\mathsf{N}[+] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m plus n equals $\mathsf{o}\}$

$\mathsf{N}[\times] = \{\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{o}\rangle \mid \mathsf{m}, \mathsf{n}, \mathsf{o} \in \mathbb{N}$, and m times n equals $\mathsf{o}\}$

Let Σ include all the sentences true on N. Now consider a language $\mathcal{L}'$ like $\mathcal{L}_{NT}^{<}$ but with the addition of a single constant c. And consider a set of sentences,

$$\Sigma' = \Sigma \cup \{\emptyset < c, S\emptyset < c, SS\emptyset < c, SSS\emptyset < c, SSSS\emptyset < c \ldots\}$$

that is like Σ but with the addition of sentences asserting that c is greater than each integer. Clearly there is no such individual on the standard interpretation N. A finite subset of Σ' can have at most finitely many of these sentences as members. Thus a finite subset of Σ' is a subset of,,

$$\Sigma \cup \{\emptyset < c, S\emptyset < c, SS\emptyset < c \ldots \overbrace{SS\ldots S}^{nS\text{'s}}\,\emptyset < c\}$$

for some n. But any such set is finitely satisfiable: Simply let the interpretation N' be like N but with $\mathsf{N}[c] = n + 1$. It follows from T11.9 that Σ' has a model M'. But, further, by reasoning as for T10.16, a model M like M' but without the assignment to c is a model of $\mathcal{L}_{NT}^{<}$ for all the sentences in Σ. So $\mathsf{N} \equiv \mathsf{M}$. But $\mathsf{N} \ncong \mathsf{M}$. For there must be a member of $\mathsf{U_M}$ with infinitely many members of $\mathsf{U_M}$ that stand in the $<$ relation to it. [Clean this up.]

It is worth observing that we have demonstrated the existence of a model for the completely nonstandard M by appeal to the more standard models M' for finite subsets of Σ', through the compactness theorem. Also, it is now clear that there can be no analog to the result of the previous section for models with an infinite domain: For models with an infinite domain, elementary equivalence does not in general imply isomorphism. In the next section, we begin to see just how general this phenomenon is.

11.5 Submodels and Löwenheim-Skolem

The construction for the adequacy theorem gives us a countable model for any consistent set of sentences. Already, this suggests that sentences for some models do not always have the same size domain. Suppose Σ has a model I. Then by T10.4, Σ is consistent; so by T10.17, Σ has a model M — where the universe of this latter model is constructed of disjoint sets of integers. But this means that if Σ has a model at all, then it has a countable model, for we might order the members of $\mathsf{U_M}$ by, say, their least elements into a countable series. In fact, we might set up a function ι from each set in $\mathsf{U_M}$ to its least element, to establish an isomorphic interpretation M^* whose universe just *is* a set of integers. Then by T11.6, $\mathsf{M}^*[\Sigma] = \mathsf{T}$. So consider any model whose universe is not countable; it must be elementarily equivalent to one whose universe is a countable set of integers. But, of course, there is no one-to-one map from an uncountable universe to a countable one, so the models are not isomorphic.

This sort of result is strengthened in an interesting way by the Löwenheim-Skolem theorems. In the first form, we show that every model has a *submodel* with a countable domain.

11.5.1 Submodels

SM A model M of a language $\mathscr{L}$ is a *submodel* of model N ($\mathsf{M} \subseteq \mathsf{N}$) iff

1. $\mathsf{U_M} \subseteq \mathsf{U_N}$,
2. For any sentence letter $\mathcal{S}$, $\mathsf{M}[\mathcal{S}] = \mathsf{N}[\mathcal{S}]$,
3. For any constant c of $\mathscr{L}$, $\mathsf{M}(c) = \mathsf{N}(c)$,
4. For any function symbol h^n of $\mathscr{L}$ and any $\langle \mathsf{a}_1 \dots \mathsf{a}_n \rangle$ from the members of $\mathsf{U_M}$, $\langle\langle \mathsf{a}_1 \dots \mathsf{a}_n \rangle, \mathsf{b}\rangle \in \mathsf{M}(h^n)$ iff $\langle\langle \mathsf{a}_1 \dots \mathsf{a}_n \rangle, \mathsf{b}\rangle \in \mathsf{N}(h^n)$,
5. For any relation symbol $\mathcal{R}^n$ of $\mathscr{L}$ and any $\langle \mathsf{a}_1 \dots \mathsf{a}_n \rangle$ from the members of $\mathsf{U_M}$, $\langle \mathsf{a}_1 \dots \mathsf{a}_n \rangle \in \mathsf{M}(\mathcal{R}^n)$ iff $\langle \mathsf{a}_1 \dots \mathsf{a}_n \rangle \in \mathsf{N}(\mathcal{R}^n)$.

The interpretation of h^n and of $\mathcal{R}^n$ on M are the *restrictions* of their respective interpretations on N. Observe that a submodel is completely determined, once its domain is given. A submodel is not well defined if it does not include objects for the interpretation of the constants, and the closure of its functions.

ES Say d is a variable assignment into the members of $\mathsf{U_M}$. Then M is an *elementary submodel* of N iff $\mathsf{M} \subseteq \mathsf{N}$ and for any formula $\mathcal{P}$ of $\mathscr{L}$ and any such d, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

If M is an elementary submodel of N, we write, $\mathsf{M} \prec \mathsf{N}$. First,

T11.10. If $\mathsf{M} \prec \mathsf{N}$ then for any sentence $\mathcal{P}$ of $\mathcal{L}$, $\mathsf{M}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{N}[\mathcal{P}] = \mathsf{T}$.

Suppose $\mathsf{M} \prec \mathsf{N}$ and consider some sentence $\mathcal{P}$. By TI, $\mathsf{M}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ for every assignment d into $\mathsf{U_M}$; since $\mathcal{P}$ is a sentence, by T8.4, iff for some particular assignment h, $\mathsf{M_h}[\mathcal{P}] = \mathsf{S}$; since $\mathsf{M} \prec \mathsf{N}$, iff $\mathsf{N_h}[\mathcal{P}] = \mathsf{S}$; since $\mathcal{P}$ is a sentence, by T8.4, iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$ for every d into $\mathsf{U_N}$; by TI, iff $\mathsf{N}[\mathcal{P}] = \mathsf{T}$. So $\mathsf{M}[\mathcal{P}] = \mathsf{T}$ iff $\mathsf{N}[\mathcal{P}] = \mathsf{T}$.

This much is clear. It is not so easy demonstrate the conditions under which a submodel is an elementary submodel. We make a beginning with the following theorems.

T11.11. Suppose $\mathsf{M} \subseteq \mathsf{N}$ and d is a variable assignment into $\mathsf{U_M}$. Then for any term t, $\mathsf{M_d}[t] = \mathsf{N_d}[t]$.

By induction on the number of function symbols in t. Suppose $\mathsf{M} \subseteq \mathsf{N}$ and d is a variable assignment into $\mathsf{U_M}$.

Basis: Suppose t has no function symbols. Then t is a variable x or a constant c. (i) Suppose t is a constant c. Then $\mathsf{M_d}[t]$ is $\mathsf{M_d}[c]$; by TA(c) this is $\mathsf{M}[c]$; and since $\mathsf{M} \subseteq \mathsf{N}$, this is $\mathsf{N}[c]$; by TA(c) again, this is $\mathsf{N_d}[c]$; which is just $\mathsf{N_d}[t]$. (ii) Suppose t is a variable x. Then $\mathsf{M_d}[t]$ is $\mathsf{M_d}[x]$; by TA(v), this is $\mathsf{d}[x]$ and by TA(v) again, this is $\mathsf{N_d}[x]$; which is just $\mathsf{N_d}[t]$.

Assp: For any i, $0 \leq i < k$, if t has i function symbols, then $\mathsf{M_d}[t] = \mathsf{N_d}[t]$.

Show: If t has k function symbols, $\mathsf{M_d}[t] = \mathsf{N_d}[t]$.

If t has k function symbols, then it is of the form $h^n t_1 \ldots t_n$ for some terms $t_1 \ldots t_n$ with $< k$ function symbols. So $\mathsf{M_d}[t]$ is $\mathsf{M_d}[h^n t_1 \ldots t_n]$; by TA(f) this is $\mathsf{M}[h^n]\langle \mathsf{M_d}[t_1], \ldots \mathsf{M_d}[t_n]\rangle$; since $\mathsf{M} \subseteq \mathsf{N}$, with the assumption, this is $\mathsf{N}[h^n]\langle \mathsf{N_d}[t_1], \ldots \mathsf{N_d}[t_n]\rangle$; by TA(f), this is $\mathsf{N_d}[h^n t_1 \ldots t_n]$; which is just $\mathsf{N_d}[t]$.

Indct: For any term t, $\mathsf{M_d}[t] = \mathsf{N_d}[t]$.

T11.12. Suppose that $\mathsf{M} \subseteq \mathsf{N}$ and that for any formula $\mathcal{P}$ and every variable assignment d such that $\mathsf{N_d}[\exists x \mathcal{P}] = \mathsf{S}$ there is an $\mathsf{m} \in \mathsf{U_M}$ such that $\mathsf{N_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$. Then $\mathsf{M} \prec \mathsf{N}$.

Suppose $\mathsf{M} \subseteq \mathsf{N}$ and that for any formula $\mathcal{P}$ and every variable assignment d such that $\mathsf{N_d}[\exists x \mathcal{P}] = \mathsf{S}$ there is an $\mathsf{m} \in \mathsf{U_M}$ such that $\mathsf{N_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$. We show by induction on the number of operators in $\mathcal{P}$, that for d any assignment into the members of $\mathsf{U_M}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

Basis: If $\mathcal{P}$ is atomic then it is either a sentence letter $\mathcal{S}$ or an atomic of the form $\mathcal{R}^n t_1 \ldots t_n$ for some relation symbol $\mathcal{R}^n$ and terms $t_1 \ldots t_n$. (i) Suppose $\mathcal{P}$ is $\mathcal{S}$. Then $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathcal{S}] = \mathsf{S}$; by SF(s), iff $\mathsf{M}[\mathcal{S}] = \mathsf{T}$; since $\mathsf{M} \subseteq \mathsf{N}$, iff $\mathsf{N}[\mathcal{S}] = \mathsf{T}$; by SF(s), iff $\mathsf{N_d}[\mathcal{S}] = \mathsf{S}$; iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$. (ii) Suppose $\mathcal{P}$ is $\mathcal{R}^n t_1 \ldots t_n$. Then $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S}$; by SF(r) iff $\langle \mathsf{M_d}[t_1], \ldots \mathsf{M_d}[t_n] \rangle \in \mathsf{M}[\mathcal{R}^n]$; since $\mathsf{M} \subseteq \mathsf{N}$ with T11.11 iff $\langle \mathsf{N_d}[t_1], \ldots \mathsf{N_d}[t_n] \rangle \in \mathsf{N}[\mathcal{R}^n]$; by SF(r) iff $\mathsf{N_d}[\mathcal{R}^n t_1 \ldots t_n] = \mathsf{S}$; iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

Assp: For any i, $0 \leq i < k$, for d any assignment into the members of $\mathsf{U_M}$, if $\mathcal{P}$ has i operator symbols, then $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

Show: If $\mathcal{P}$ has k operator symbols, then for d any assignment into the members of $\mathsf{U_M}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

If $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \rightarrow \mathcal{B}$ or $\exists x \mathcal{A}$ for variable x and formulas A and $\mathcal{B}$ with $< k$ operator symbols (treating universally quantified expressions as equivalent to existentially quantified ones). Let d be an assignment into the members of $\mathsf{U_M}$.

($\sim$) Suppose $\mathcal{P}$ is $\sim A$. $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{M_d}[\sim\mathcal{A}] = \mathsf{S}$; by SF($\sim$) iff $\mathsf{M_d}[\mathcal{A}] \neq \mathsf{S}$; by assumption iff $\mathsf{N_d}[\mathcal{A}] \neq \mathsf{S}$; by SF($\sim$) iff $\mathsf{N_d}[\sim\mathcal{A}] = \mathsf{S}$; iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

($\rightarrow$) Homework.

($\exists$) Suppose $\mathcal{P}$ is $\exists x \mathcal{A}$. (i) Suppose $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$; then $\mathsf{M_d}[\exists x \mathcal{A}] = \mathsf{S}$; so by SF($\exists$), there is some $\mathsf{o} \in \mathsf{U_M}$ such that $\mathsf{M_{d(x|o)}}[\mathcal{A}] = \mathsf{S}$; so since $\mathsf{d}(x|\mathsf{o})$ is an assignment into the members of $\mathsf{U_M}$, by assumption, $\mathsf{N_{d(x|o)}}[\mathcal{A}] = \mathsf{S}$; so by SF($\exists$), $\mathsf{N_d}[\exists x \mathcal{A}] = \mathsf{S}$; so $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$. (ii) Suppose $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$; then $\mathsf{N_d}[\exists x \mathcal{A}] = \mathsf{S}$; so by the assumption of the theorem, there is an $\mathsf{m} \in \mathsf{U_M}$ such that $\mathsf{N_{d(x|m)}}[\mathcal{A}] = \mathsf{S}$; since $\mathsf{d}(x|\mathsf{m})$ is an assignment into the members of $\mathsf{U_M}$, by assumption $\mathsf{M_{d(x|m)}}[\mathcal{A}] = \mathsf{S}$; so by SF($\exists$), $\mathsf{M_d}[\exists x \mathcal{A}] = \mathsf{S}$; so $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$. So $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

In any case, if $\mathcal{P}$ has k operator symbols, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

Indct: For any $\mathcal{P}$, $\mathsf{M_d}[\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\mathcal{P}] = \mathsf{S}$.

So the result works, only so long as the quantifier case is guaranteed by "witnesses" for each existential claim in the universe of the submodel. The Löwenheim Skolem Theorem takes advantage of what we have done by producing a model in which these witnesses are present.

11.5.2 Downward Löwenheim-Skolem

The Löwenheim Skolem Theorem takes advantage of what we have just done by producing a model in which the required witnesses are present.

$\mathsf{U_M}$ Consider some model N and suppose a well-ordering of the objects of $\mathsf{U_N}$. We construct a countable submodel M as follows. Let A_0 be a countable subset of $\mathsf{U_N}$. We construct a series A_0, A_1, A_2 .... For a formula of the form $\exists x\mathcal{P}$ in the language $\mathcal{L}$, and a variable assignment d into A_i, let d' be like d for the initial segment that assigns to variables free in $\mathcal{P}$, and after assigns to a constant object m_0 in A_0. Then for any $\mathcal{P}$ and d such that $\mathsf{N_d}[\exists x\mathcal{P}] = \mathsf{S}$, find the first object o in the well-ordering of $\mathsf{U_N}$ such that $\mathsf{N}_{\mathsf{d}'(x|\mathsf{o})}[\mathcal{P}] = \mathsf{S}$. To form A_{i+1}, augment A_i with all the objects obtained this way. Because there are countably many formulas, and countably many initial segments of the variable assignments, countably many objects are added to form A_{i+1}, and if A_i is countable, A_{i+1} is countable. Let $\mathsf{U_M}$ be $\bigcup_{i\geq 0}\mathsf{A}_i$. Again, if each A_i is countable, $\mathsf{U_M}$ is countable.

There may be uncountably many variable assignments into a given A_i. However, for a given formula $\mathcal{P}$, no matter how may assignments there may be on which it is satisfied, there can be at most countably many initial segments of the sort d'. So at most countably many objects are added. The functions from formulas and variable assignments to individuals are *Skolem* functions, and we consider the closure of A under the set of all Skolem functions.

T11.13. With $\mathsf{U_M}$ constructed as above, a submodel M of N is well-defined.

Clearly $\mathsf{U_M} \subseteq \mathsf{U_N}$. For constants, consider the case when $\exists x\mathcal{P}$ is $\exists x(x = c)$; then at any stage i, $\mathsf{M}_{\mathsf{d}'(x|\mathsf{o})}[x = c] = \mathsf{S}$ iff $\mathsf{o} = \mathsf{M}[c]$. So $\mathsf{M}[c]$ is a member of A_{i+1} and so of $\mathsf{U_M}$. Similarly, for functions, consider the case when $\exists x\mathcal{P}$ is $\exists x(h^n v_1 \ldots v_n = x)$ for some function symbol h^n and variables $v_1 \ldots v_n$ and x. For any d, consider some d' which assigns objects to each of the variables $v_1 \ldots v_n$; then there there is some A_i such that d' is an assignment into it; so by construction, A_{i+1} includes an object o

such that $\mathsf{N}_{\mathsf{d}'(x|\mathsf{o})}[\mathscr{h}^n v_1 \ldots v_n = x] = \mathsf{S}$. But this must be the object $\mathsf{N}[\mathscr{h}^n]\langle \mathsf{N}_{\mathsf{d}'}[v_1], \ldots \mathsf{N}_{\mathsf{d}'}[v_1]\rangle$.

T11.14. For any model N there is an $\mathsf{M} \prec \mathsf{N}$ such that M has a countable domain. (*Löwenheim-Skolem*)

To show $\mathsf{M} \prec \mathsf{N}$ by T11.12, it remains to show that for any formula $\mathscr{P}$ and every variable assignment d such that $\mathsf{N}_{\mathsf{d}}[\exists x \mathscr{P}] = \mathsf{S}$ there is an $\mathsf{m} \in \mathsf{U}_{\mathsf{M}}$ such that $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathscr{P}] = \mathsf{S}$. But this is easy. Suppose $\mathsf{N}_{\mathsf{d}}[\exists x \mathscr{P}] = \mathsf{S}$; then where d and d' agree on assignments to all the free variables in $\mathscr{P}$, by T8.4, $\mathsf{N}_{\mathsf{d}'}[\exists x \mathscr{P}] = \mathsf{S}$. But all assignments from d' are elements of some A_i; so by construction there is object m such that $\mathsf{N}_{\mathsf{d}'(x|\mathsf{m})}[\mathscr{P}] = \mathsf{S}$ in A_{i+1} and so in U_{M}; and since d and d' agree on their assignments to all the free variables in $\mathscr{P}$, by T8.4, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathscr{P}] = \mathsf{S}$.

[applications]

11.5.3 Upward Löwenheim-Skolem

Part IV

Logic and Arithmetic: Incompleteness and Computability

Introductory

In Part III we showed that our semantical and syntactical logical notions are related as we want them to be: exactly the same arguments are semantically valid as are provable. So,

$$\Gamma \vdash \mathcal{P} \quad \text{iff} \quad \Gamma \vDash \mathcal{P}$$

Thus our derivation system is both sound and adequate, as it should be. In this part, however, we encounter a series of limiting results — with particular application to arithmetic and computing.

First, it is natural to think of mathematics as characterized by proofs and derivations. Thus, one might anticipate that there would be some system of premises Δ such that for any $\mathcal{P}$ in $\mathcal{L}_{\text{NT}}$, with N the standard interpretation of number theory, we would have,

$$\Delta \vdash \mathcal{P} \quad \text{iff} \quad \mathsf{N}[\mathcal{P}] = \mathsf{T}$$

Note the difference between our claims. In the first, derivations from premises are matched to entailments from premises; in the second, derivations (and so entailments) are matched to truths on an interpretation. Perhaps inspired by suspicions about the existence or nature of numbers, one might expect that derivations would even entirely replace the notion of mathematical truth. And Q or PA may already seem to be deductive systems of this sort. But we shall see that there can be no such deductive system. From Gödel's first incompleteness theorem, under certain constraints, no consistent deductive system has as consequences either $\mathcal{P}$ or $\sim\mathcal{P}$ for every $\mathcal{P}$ of $\mathcal{L}_{\text{NT}}$; any such theory is (negation) *incomplete*. But then, subject to those constraints, any consistent deductive system must omit some truths of arithmetic from among its consequences.[2]

[2] Gödel's groundbreaking paper is "On the Formally Undecidable Propositions of *Principia Mathematica* and Related Systems."

Suppose there is no one-to-one map between truths of arithmetic and consequences of our theories. Rather, we propose a theory R(eal) whose consequences are unproblematically true, and another theory I(deal) whose consequences outrun those of R and whose literal truth is therefore somehow suspect. Perhaps R is sufficient only for something like basic arithmetic, whereas I seems to quantify over all members of a far-flung infinite domain. Even though not itself a vehicle for truth, theory I may be useful under certain circumstances. Suppose,

(a) For any $\mathscr{P}$ in the scope of R, if $\mathscr{P}$ is not true, then $R \vdash \sim\mathscr{P}$

(b) I *extends* R: If $R \vdash \mathscr{P}$ then $I \vdash \mathscr{P}$

(c) I is *consistent*: There is no $\mathscr{P}$ such that $I \vdash \mathscr{P}$ and $I \vdash \sim\mathscr{P}$

Then theory I may be treated as a tool for achieving results in the scope of R: Suppose $\mathscr{P}$ is a result in the scope of R, and $I \vdash \mathscr{P}$; then by consistency, $I \nvdash \sim\mathscr{P}$; and because I extends R, $R \nvdash \sim\mathscr{P}$; so by (a), $\mathscr{P}$ is true. This is (a sketch of) the famous 'Hilbert program' for mathematics, which aims to make sense of infinitary mathematics based not on the truth but rather the consistency of theory I.

Because consistency is a syntactical result about proof systems, not itself about far-flung mathematical structures, one might have hoped for proofs of consistency from real, rather than ideal, theories. But Gödel's second incompleteness theorem tells us that derivation systems extending PA cannot prove even their own consistency. So a weaker "real" theory will not be able to prove the consistency of PA and its extensions. But this seems to remove a demonstration of (c) and so to doom the Hilbert strategy.[3]

Even though no one derivation system has as consequences every mathematical truth, derivations remain useful, and mathematicians continue to do proofs! Given that we care about them, there is a question about the automation of proofs. Say a property or relation is *effectively decidable* iff there is an algorithm or program that

[3] We are familiar with the Pythagorean Theorem according to which the hypotenuse and sides of a right triangle are such that $a^2 = b^2 + c^2$. In the 1600s Fermat famously proposed that there are no integers a, b, c such that $a^n = b^n + c^n$ for $n > 2$; so, for example, there are no a, b, c such that $a^3 = b^3 + c^3$. In 1995 Andrew Wiles proved that this is so. But Wiles's proof requires some fantastically abstract (and difficult) mathematics. Even if Wiles's abstract theory (I) is not *true* Hilbert could still accept the demonstration of Fermat's (real) theorem so long as I is shown to be *consistent*. Gödel's result seems to doom this strategy. Of course, one might simply accept Wiles's proof on the ground that his advanced mathematics is *true* so that its consequences are true as well. But this is a topic in philosophy of mathematics, not logic! See, for example, Shapiro, *Thinking About Mathematics* for an introduction to options in the philosophy of mathematics. Our limiting results may very well stimulate interest in that field!

could decide in a finite number of steps whether the property or relation applies in any given case. Abstracting from the limitations of particular computing devices, we shall identify a class of relations which are decidable. A corollary of Gödel's first theorem is that validity in systems like *ND* and *AD* is not among the decidable relations. Thus there are interesting limits on the decidable relations — where it is possible also to look back through this lense at Gödel's first theorem.

Chapter 12 lays down background required for chapters that follow. It begins with a discussion of *recursive functions*, and concludes with a few essential results, including a demonstration of the incompleteness of arithmetic. Chapters 13 and 14 deepen and extend those results in different ways. Chapter 13 includes Gödel's own argument for incompleteness from the construction of a sentence such that neither it nor its negation is provable, along with demonstration of the second incompleteness theorem. Chapter 14 again shows that there must exist a sentence such that neither it nor its negation is provable, but this time in association with an account of computability. Chapter 12 is required for either chapter 13 or chapter 14; but those chapters may be taken in either order.

Chapter 12

Recursive Functions and Q

A formal *theory* consists of a language, with some axioms and proof system. Q and PA are example theories. A theory T is (negation) *complete* iff for any sentence $\mathcal{P}$ in its language $\mathcal{L}$, either $T \vdash \mathcal{P}$ or $T \vdash \sim\mathcal{P}$. Observe again that a derivation system is adequate when it proves every entailment of some premises. Our standard logic does that. Granting then, the adequacy of the logic, negation completeness is a matter of premises proving a sufficiently robust set of consequences — proving consequences which include $\mathcal{P}$ or $\sim\mathcal{P}$ for every $\mathcal{P}$ in the language.

Let us pause to consider why completeness matters. From E8.22, as soon as a language $\mathcal{L}$ has an interpretation I, for any sentence $\mathcal{P}$ in $\mathcal{L}$, either $\mathsf{I}[\mathcal{P}] = \mathsf{T}$ or $\mathsf{I}[\sim\mathcal{P}] = \mathsf{T}$. So if we set out to characterize by means of a theory the sentences that are true on some interpretation, our theory is bound to omit some sentences unless it is such that for any $\mathcal{P}$, either $T \vdash \mathcal{P}$ or $T \vdash \sim\mathcal{P}$. To the extent that we desire a characterization of all true sentences in some domain, of arithmetic or whatever, a complete theory is a desirable theory.[1]

By itself negation completeness is no extraordinary thing. Consider a theory whose language has just two sentence letters A and B, along with the usual sentential operators and rules. The axioms of our theory are just A and $\sim B$. On a truth table, there is just one row were these axioms are both true, and on that row, any $\mathcal{P}$ in the language is either T or F, so that one of $\mathcal{P}$ or $\sim\mathcal{P}$ is T.

[1] We thus restrict ourselves to consideration of *sentences* as theorems — or, equivalently treat open formulas as equivalent to their universal closures (see p. 486)

(A)

A	B	A	$\sim B$	/	$\mathscr{P}$	$\sim\mathscr{P}$
T	T	***T***	***F***		–	–
T	F	***T***	***T***		***T/F***	***F/T***
F	T	***F***	***F***		–	–
F	F	***F***	***T***		–	–

So for any $\mathscr{P}$, either $A, \sim B \vDash \mathscr{P}$ or $A, \sim B \vDash \sim\mathscr{P}$. But from the adequacy of the derivation system if $\Gamma \vDash \mathscr{P}$, then $\Gamma \vdash \mathscr{P}$ (T10.11, p. 482); so for any $\mathscr{P}$, either $A, \sim B \vdash \mathscr{P}$ or $A, \sim B \vdash \sim\mathscr{P}$. So our little theory with its restricted language is negation complete. Contrast this with a theory that has the same language and rules, but A as its only axiom. In this case, it is easy to see from truth tables that, say, $A \nvDash B$ and $A \nvDash \sim B$. But by soundness, if $\Gamma \vdash \mathscr{P}$ then $\Gamma \vDash \mathscr{P}$ (T10.3, p. 468); it follows that $A \nvdash B$ and $A \nvdash \sim B$. So this theory is not negation complete.

These theories are not very interesting. However, let $\mathscr{L}_{NT}^{S+}$ be a language like $\mathscr{L}_{NT}$ whose only function symbols are S and $+$ (without $\times$), and let $\mathscr{L}_{NT}^{\times}$ be a language like $\mathscr{L}_{NT}$ whose only function symbol is $\times$ (without S and $+$). Then there is a complete theory for the arithmetic of $\mathscr{L}_{NT}^{S+}$ (*Presburger Arithmetic*), and a complete theory for the arithmetic of $\mathscr{L}_{NT}^{\times}$ (*Skolem Arithmetic*).[2] These are interesting and powerful theories. So, again, by itself negation completeness is not so extraordinary.

However there is no complete theory for the arithmetic of $\mathscr{L}_{NT}$ which includes all of S, $+$ and $\times$. It turns out that theories are something like superheros. In the ordinary case, a complete, and so a "happy" life is at least within reach. However, as theories acquire certain powers, they take on a "fatal flaw" just because of their powers — where this flaw makes completeness unattainable. On its face, theory Q does not appear particularly heroic. We have seen already in E7.20 that Q $\nvdash x \times y = y \times x$ and Q $\nvdash \sim(x \times y = y \times x)$. So Q is negation incomplete. PA which does prove $x \times y = y \times x$ along with other standard results in arithmetic might seem a more likely candidate for heroism. But Q includes already features sufficient to generate the flaw which appears also in any theories, like PA, which have at least all the powers of Q. It is our task to identify this flaw.

It turns out that a system with the powers of Q including S, $+$ and $\times$ can express and capture all the *recursive* functions — and a system with these powers must have the fatal flaw. Thus, in this chapter we focus on the recursive functions, and associate them with powers of our formal systems. We conclude with a few applications from these powers.

[2]For demonstration of completeness for Presburger Arithmetic, see Fisher, *Formal Number Theory and Computability* chapter 7 along with Boolos, Burgess and Jeffrey, *Computability and Logic* chapter 24.

12.1 Recursive Functions

In chapter 6 (p. 314) for Q and PA we had axioms of the sort,

a. $x + \emptyset = x$
b. $x + Sy = S(x + y)$

and

c. $x \times \emptyset = \emptyset$
d. $x \times Sy = (x \times y) + x$

These enable us to derive $x + y$ and $x \times y$ for arbitrary values of x and y. Thus, by (a) $2 + 0 = 2$; so by (b) $2 + 1 = 3$; and by (b) again, $2 + 2 = 4$; and so forth. From the values at any one stage, we are in a position to calculate values at the next. And similarly for multiplication. From E6.35 on p. 315, all this should be familiar.

While axioms thus supply effective means for calculating the values of these functions, the functions themselves might be similarly *identified* or *specified*. So, given a successor function $\mathsf{suc}(\mathsf{x})$, we may identify the functions $\mathsf{plus}(\mathsf{x}, \mathsf{y})$:

a. $\mathsf{plus}(\mathsf{x}, 0) = \mathsf{x}$
b. $\mathsf{plus}(\mathsf{x}, \mathsf{suc}(\mathsf{y})) = \mathsf{suc}(\mathsf{plus}(\mathsf{x}, \mathsf{y}))$

and $\mathsf{times}(\mathsf{x}, \mathsf{y})$:

c. $\mathsf{times}(\mathsf{x}, 0) = 0$
d. $\mathsf{times}(\mathsf{x}, \mathsf{suc}(\mathsf{y})) = \mathsf{plus}(\mathsf{times}(\mathsf{x}, \mathsf{y}), \mathsf{x})$

For ease of reading, let us typically revert to the more ordinary notation S, $\mathbf{+}$ and $\mathbf{\times}$ for these functions, though we stick with the (emphasized) sans serif font. We have been thinking of functions as certain complex sets. Thus the plus function is a set with elements $\{\ldots \langle\langle 2, 0\rangle, 2\rangle, \langle\langle 2, 1\rangle, 3\rangle, \langle\langle 2, 2\rangle, 4\rangle \ldots\}$. Our specification picks out this set. From the first clause, $\mathsf{plus}(\mathsf{x}, \mathsf{y})$ has $\langle\langle 2, 0\rangle, 2\rangle$ as a member; given this, $\langle\langle 2, 1\rangle, 3\rangle$ is a member; and so forth. So the two clauses work together to specify the plus function. And similarly for times.

But these are not the only sets which may be specified this way. Thus the standard factorial $\mathsf{fact}(\mathsf{x})$:

e. $\mathsf{fact}(0) = \mathsf{S}(0)$
f. $\mathsf{fact}(\mathsf{Sy}) = \mathsf{fact}(\mathsf{y}) \times \mathsf{Sy}$

Again, we will often revert to the more typical $\mathsf{x}\mathbf{!}$ notation. Zero factorial is one. And the factorial of Sy multiplies $1 \times 2 \times \ldots \times \mathsf{y}$ by Sy. Similarly $\mathsf{power}(\mathsf{x}, \mathsf{y})$:

g. $\mathsf{power}(\mathsf{x}, 0) = \mathsf{S0}$
h. $\mathsf{power}(\mathsf{x}, \mathsf{Sy}) = \mathsf{power}(\mathsf{x}, \mathsf{y}) \times \mathsf{x}$

Any number to the power of zero is one ($x^0 = 1$). And then x^{Sy} multiplies $x^y = x \times x \ldots \times x$ (y times) by another x.

We shall be interested in a class of functions, the *recursive* functions, which may be specified (in part) by this two-stage strategy. To make progress, we turn to a general account in five stages.

12.1.1 Initial Functions

Our examples have simply taken $\mathsf{suc}(x)$ as given. Similarly, we shall require a stock of *initial functions*. There are initial functions of three different types.

First, we shall continue to include $\mathsf{suc}(x)$ among the initial functions. So $\mathsf{suc}(x) = \{\langle 0,1\rangle, \langle 1,2\rangle, \langle 2,3\rangle \ldots\}$.

Second, $\mathsf{zero}(x)$ is a function which returns zero for any input value. So $\mathsf{zero}(x) = \{\langle 0,0\rangle, \langle 1,0\rangle, \langle 2,0\rangle \ldots\}$.

Finally, for any $1 \leq k \leq j$, we require a collection of *identity* functions $\mathsf{idnt}^j_k(x_1 \ldots x_j)$. Each idnt^j_k function has j places and simply returns the value from the k^{th} place. Thus $\mathsf{idnt}^3_2(4,5,6) = 5$. So, $\mathsf{idnt}^3_2 = \{\ldots \langle\langle 1,2,3\rangle,2\rangle \ldots \langle\langle 4,5,6\rangle,5\rangle \ldots\}$. And in the simplest case, $\mathsf{idnt}^1_1(x) = x$.

12.1.2 Composition

In our examples, we have let one function be *composed* from others — as when we consider $\mathsf{times}(x, \mathsf{suc}(y))$ or the like. Say $\vec{x}$ represents a (possibly empty) series of variables $x_1 \ldots x_n$.

CM Let $g(\vec{y})$ and $h(\vec{x}, w, \vec{z})$ be any functions. Then $f(\vec{x}, \vec{y}, \vec{z})$ is defined by *composition* from $g(\vec{y})$ and $h(\vec{x}, w, \vec{z})$ iff $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$.

So $h(\vec{x}, w, \vec{z})$ gets its value in the w-place from $g(\vec{y})$. Here is a simple example: $f(y,z) = \mathsf{zero}(y) + z$ results by composition from substitution of $\mathsf{zero}(y)$ into $\mathsf{plus}(w,z)$; so $\mathsf{plus}(w,z)$ gets its value in the w-place from $\mathsf{zero}(y)$. The result is the set with members, $\{\ldots \langle\langle 2,0\rangle,0\rangle, \langle\langle 2,1\rangle,1\rangle, \langle\langle 2,2\rangle,2\rangle \ldots\}$. Given, say, input $\langle 2,2\rangle$, $\mathsf{zero}(y)$ takes the input 2 and supplies a zero to the first place of the $\mathsf{plus}(x,y)$ function; then from $\mathsf{plus}(x,y)$ the result is a sum of 0 and 2 which is 2. And similarly in other cases. In contrast, $\mathsf{zero}(x+y)$ has members $\{\ldots \langle\langle 2,0\rangle,0\rangle, \langle\langle 2,1\rangle,0\rangle, \langle\langle 2,2\rangle,0\rangle \ldots\}$. You should see how this works.

12.1.3 Recursion

For each of our examples, $\mathsf{plus}(x, y)$, $\mathsf{times}(x, y)$, $\mathsf{fact}(y)$, and $\mathsf{power}(x, y)$, the value of the function is set for $y = 0$ and then for $\mathsf{suc}(y)$ given its value for y. These illustrate the method of recursion. Put generally,

RC Given some functions $g(\vec{x})$ and $h(\vec{x}, y, u)$, $f(\vec{x}, y)$ is defined by *recursion* when,

$$f(\vec{x}, 0) = g(\vec{x})$$
$$f(\vec{x}, Sy) = h(\vec{x}, y, f(\vec{x}, y))$$

This general scheme includes flexibility that is not always required. In the cases of plus, times and power, $\vec{x}$ reduces to a simple variable x; for fact, $\vec{x}$ disappears altogether, so that the function $g(\vec{x})$ reduces to a constant. And, as we shall see, the function $h(\vec{x}, y, u)$ need not depend on each of its variables x, y and u.

However, by clever use of our initial functions, it is possible to see each of our sample functions on this pattern. Thus for $\mathsf{plus}(x, y)$, set $\mathsf{gplus}(x) = \mathsf{idnt}^1_1(x)$ and $\mathsf{hplus}(x, y, u) = \mathsf{suc}(\mathsf{idnt}^3_3(x, y, u))$. Then,

a′ $\mathsf{plus}(x, 0) = \mathsf{idnt}^1_1(x)$

b′ $\mathsf{plus}(x, Sy) = \mathsf{suc}(\mathsf{idnt}^3_3(x, y, \mathsf{plus}(x, y)))$

And these work as they should: $\mathsf{idnt}^1_1(x) = x$ and $\mathsf{suc}(\mathsf{idnt}^3_3(x, y, \mathsf{plus}(x, y)))$ is the same as $\mathsf{suc}(\mathsf{plus}(x, y))$. So we recover the conditions (a) and (b) from above.

Similarly, for $\mathsf{times}(x, y)$, let $\mathsf{gtimes}(x) = \mathsf{zero}(x)$ and $\mathsf{htimes}(x, y, u) = \mathsf{plus}(\mathsf{idnt}^3_3(x, y, u), x)$. Then,

c′ $\mathsf{times}(x, 0) = \mathsf{zero}(x)$

d′ $\mathsf{times}(x, Sy) = \mathsf{plus}(\mathsf{idnt}^3_3(x, y, \mathsf{times}(x, y)), x)$

So $\mathsf{times}(x, 0) = 0$ and $\mathsf{times}(x, Sy) = \mathsf{plus}(\mathsf{times}(x, y), x))$, and all is well. Observe that we would obtain the same result with $\mathsf{htimes}(x, y, u) = \mathsf{plus}(u, \mathsf{idnt}^3_1(x, y, u))$ or perhaps, $\mathsf{plus}(\mathsf{idnt}^3_3(x, y, u), \mathsf{idnt}^3_1(x, y, u))$. The role of the identity functions in these formulations is to preserve h as a function of x, y and u, even where not each place is required — as the y-place is not required for times, and so to adhere to the official form which makes $h(x, y, u)$ a function of variables in each place. And there are these different ways to produce a function of all the variables to achieve the desired result.

In the case of $\mathsf{fact}(y)$, there are no places to the $\vec{x}$ vector. So gfact is reduced to a zero-place function, that is, to a constant, and hfact to a function of y and u. In contrast, for $\mathsf{times}(x, y)$, $\vec{x}$ retains one place, so $\mathsf{gtimes}(x)$ is not reduced to a constant; rather $\mathsf{gtimes}(x) = \mathsf{zero}(x)$ remains a full-fledged function — only one

which returns the same value for every value of x. For $\mathsf{fact(y)}$, set $\mathsf{gfact} = \mathsf{suc(0)}$ and $\mathsf{hfact(y,u)} = \mathsf{times(u,suc(y))}$. Again, identity functions work to preserve h as a function y, and u, even where not each place is required, in order to adhere to the official form. However, there is no requirement that the places be picked out by identity functions! In this case, each variable is used in a natural way, so identity functions are not required. It is left as an exercise to show that gfact and hfact identify the same function as constraints (e), (f), and to then to find $\mathsf{gpower(x)}$ and $\mathsf{hpower(x,y,u)}$.

12.1.4 Regular Minimization

So far, the method of our examples is easily matched to the capacities of computing devices. To find the value of a recursive function, begin by finding values for $\mathsf{y} = \mathsf{0}$, and then calculate other values, from one stage to the next. But this is just what computing devices do well. So, for example, in the syntax of the Ruby language,[3] given some functions `g(x)` and `h(x,y,u)`,

(B)

```
def recfunc(a,b)
 k = g(a)
 for y in 0..b-1
  k = h(a,y,k)
 end
 return k
end
```

Using `g(a)` this program calculates the value of `k` for input `(a,0)`. And then, given the current value of `y`, and of `k` for input `(a,y)`, repeatedly uses `h` to calculate `k` for the next value of `y`, until it finally reaches and returns the value of `k` for input `(a,b)`. Observe that the calculation of `recfunc(a,b)` requires exactly `b` iterations before it completes.

But there is a different repetitive mechanism available for computing devices — where this mechanism does not begin with a fixed number of iterations. Suppose we have some function `g(a,b)` with values `g(a,0)`, `g(a,1)`, `g(a,2)`... where for each `a` there are at least some values of `b` such that `g(a,b) = 0`. For any value of `a`, suppose we want the least `b` such that `g(a,b) = 0`. Then we might reason as follows.

[3] Ruby is convenient insofar as it is interpreted and so easy to run, and available at no cost on multiple platforms (see http://www.ruby-lang.org/en/downloads/). We depend only on very basic features familiar from most any exposure to computing.

The Recursion Theorem

One may wonder whether our specification $\mathsf{f}(\mathsf{x},\mathsf{y})$ by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{u})$ results in a unique function. However it is possible to show that it does.

RT Suppose $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{u})$ are total functions on $\mathbb{N}$; then there exists a unique function $\mathsf{f}(\vec{\mathsf{x}},\mathsf{y})$ such that for any $\vec{\mathsf{x}}$ and $\mathsf{y} \in \omega$,

a. $\mathsf{f}(\vec{\mathsf{x}},0) = \mathsf{g}(\vec{\mathsf{x}})$

b. $\mathsf{f}(\vec{\mathsf{x}},\mathsf{suc}(\mathsf{y})) = \mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{f}(\vec{\mathsf{x}},\mathsf{y}))$

We identify this function as a union of functions which may be constructed by means of g and h. The *domain* of a total function from r^{n} to s is always r^{n}; for a partial function, the domain of the function is that subset of r^{n} whose members are matched by the function to members of s (for background see the set theory reference p. 114). Say a (maybe partial) function $\mathsf{s}(\vec{\mathsf{x}},\mathsf{y})$ is *acceptable* iff,

i. If $\langle\vec{\mathsf{x}},0\rangle \in \mathsf{dom}(\mathsf{s})$, then $\mathsf{s}(\vec{\mathsf{x}},0) = \mathsf{g}(\vec{\mathsf{x}})$

ii. If $\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{n})\rangle \in \mathsf{dom}(\mathsf{s})$, then $\langle\vec{\mathsf{x}},\mathsf{n}\rangle \in \mathsf{dom}(\mathsf{s})$ and $\mathsf{s}(\vec{\mathsf{x}},\mathsf{suc}(\mathsf{n})) = \mathsf{h}(\vec{\mathsf{x}},\mathsf{n},\mathsf{s}(\vec{\mathsf{x}},\mathsf{n}))$

A function with members $\{\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{g}(\vec{\mathsf{x}})\rangle,\langle\langle\vec{\mathsf{x}},1\rangle,\mathsf{h}(\vec{\mathsf{x}},0,\mathsf{g}(\vec{\mathsf{x}}))\rangle\}$ would satisfy (i) and (ii). A function which satisfies the theorem is acceptable, though not every function which is acceptable satisfies the theorem; we show just one acceptable function satisfies the theorem. Let F be the collection of all acceptable functions, and f be $\bigcup\mathsf{F}$. Thus $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle \in \mathsf{f}$ iff $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle$ is a member of some acceptable s; iff $\mathsf{s}(\vec{\mathsf{x}},\mathsf{n}) = \mathsf{a}$ for some acceptable s. We sketch reasoning to show that f has the right features.

I. If $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{b}\rangle \in \mathsf{s}'$, then $\mathsf{a} = \mathsf{b}$. By induction on n: Suppose $\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{a}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{b}\rangle \in \mathsf{s}'$; then by (i), $\mathsf{a} = \mathsf{b} = \mathsf{g}(\vec{\mathsf{x}})$. Assume that if $\langle\langle\vec{\mathsf{x}},\mathsf{k}\rangle,\mathsf{a}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{k}\rangle,\mathsf{b}\rangle \in \mathsf{s}'$ then $\mathsf{a} = \mathsf{b}$. Show that if $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{c}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{d}\rangle \in \mathsf{s}'$ then $\mathsf{c} = \mathsf{d}$. So suppose $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{c}\rangle \in \mathsf{s}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{d}\rangle \in \mathsf{s}'$. Then by (ii) $\mathsf{c} = \mathsf{h}(\vec{\mathsf{x}},\mathsf{k},\mathsf{s}(\vec{\mathsf{x}},\mathsf{k}))$ and $\mathsf{d} = \mathsf{h}(\vec{\mathsf{x}},\mathsf{k},\mathsf{s}'(\vec{\mathsf{x}},\mathsf{k}))$. But by by assumption $\mathsf{s}(\vec{\mathsf{x}},\mathsf{k}) = \mathsf{s}'(\vec{\mathsf{x}},\mathsf{k})$; so $\mathsf{c} = \mathsf{d}$.

II. $\mathsf{dom}(\mathsf{f})$ includes every $\langle\vec{\mathsf{x}},\mathsf{n}\rangle$. By induction on n: For any $\vec{\mathsf{x}}$, $\{\langle\langle\vec{\mathsf{x}},0\rangle,\mathsf{g}(\vec{\mathsf{x}})\rangle\}$ is itself an acceptable function. Assume that for any $\vec{\mathsf{x}}$, $\langle\vec{\mathsf{x}},\mathsf{k}\rangle \in \mathsf{dom}(\mathsf{f})$. Show that for any $\vec{\mathsf{x}}$, $\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle \in \mathsf{dom}(\mathsf{f})$. Suppose otherwise, and consider a function, $\mathsf{s} = \mathsf{f} \cup \{\langle\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle,\mathsf{h}(\vec{\mathsf{x}},\mathsf{k},\mathsf{f}(\vec{\mathsf{x}},\mathsf{k}))\rangle\}$. But we may show that s so defined is an acceptable function; and since s is acceptable, it is a subset of f; so $\langle\vec{\mathsf{x}},\mathsf{suc}(\mathsf{k})\rangle \in \mathsf{dom}(\mathsf{f})$. Reject the assumption.

III. Now by (I), if $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{a}\rangle \in \mathsf{f}$ and $\langle\langle\vec{\mathsf{x}},\mathsf{n}\rangle,\mathsf{b}\rangle \in \mathsf{f}$, then $\mathsf{a} = \mathsf{b}$; so f is a function; and by (II) the domain of f is all of ω; by construction it is easy to see that f is itself acceptable. From this, f satisfies the theorem. With (I), f is the unique acceptable function which satisfies the theorem; and since any function that satisfies the theorem is acceptable, the theorem is uniquely satisfied.

*We employ *weak* induction from the induction schemes reference p. 384. Enderton, *Elements of Set Theory*, and Drake and Singh, *Intermediate Set Theory*, include nice discussions of this result.

(C)

```
def minfunc(a)
  y = 0
  until g(a,y) == 0
    y = y+1
  end
  return y
end
```

This program begins with `y = 0` and tests each value of `g(a,y)` until it returns a value of `0`. Once it finds this value, `minfunc(a)` is set equal to `y`. Given `g(a,b)`, then, `minfunc(a)` calculates a function which returns some value of `y` for any input value a.

But, as before, we might reason similarly to *specify* functions so calculated. For this, recall that a function is *total* iff it is defined on all members of its domain. Say a function $g(\vec{x}, y)$ is *regular* iff it is total and for all values of $\vec{x}$ there is at least one y such that $g(\vec{x}, y) = 0$. Then,

RM If $g(\vec{x}, y)$ is a regular function, the function $f(\vec{x}) = \mu y[g(\vec{x}, y)]$ which for each $\vec{x}$ takes as its value the least y such that $g(\vec{x}, y) = 0$ is defined by *regular minimization* from $g(\vec{x}, y)$.

For a simple example, consider a domain which consists of nonempty sets of integers with a function such that $g(x, y) = 0$ iff $y \in x$. Then $f(x) = \mu y[g(x, y)]$ is the least element of x.

12.1.5 Final Definition

Finally, our sample functions are *cumulative*. Thus $plus(x, y)$ depends on $suc(x)$; $times(x, y)$, on $plus(x, y)$, and so forth. We are thus led to our final account.

RF A function f_k is *recursive* iff there is a series of functions $f_0, f_1 \dots f_k$ such that for any $i \leq k$,

(i) f_i is an initial function $suc(x)$, $zero(x)$ or $idnt^j_k(x_1 \dots x_j)$.

(c) There are $a, b < i$ such that $f_i(\vec{x}, \vec{y}, \vec{z})$ results by composition from $f_a(\vec{y})$ and $f_b(\vec{x}, w, \vec{z})$.

(r) There are $a, b < i$ such that $f_i(\vec{x}, y)$ results by recursion from $f_a(\vec{x})$ and $f_b(\vec{x}, y, u)$.

(m) There is some $a < i$ such that $f_i(\vec{x})$ results by regular minimization from $f_a(\vec{x}, y)$.

If there is a series of functions f_0, $\mathsf{f}_1 \ldots$ f_k such that for any $\mathsf{i} \leq \mathsf{k}$, just (i), (c) or (r), then (PR) f_k is *primitive recursive*.

So any recursive function results from a series of functions each of which satisfies one of these conditions. And such a series demonstrates that its members are recursive. For a simple example, plus is primitive recursive.

(D)

1.	$\mathsf{idnt}^1_1(x)$	initial function
2.	$\mathsf{idnt}^3_3(x, y, u)$	initial function
3.	$\mathsf{suc}(w)$	initial function
4.	$\mathsf{suc}(\mathsf{idnt}^3_3(x, y, u))$	2,3 composition
5.	$\mathsf{plus}(x, y)$	1,4 recursion

From this list by itself, one might reasonably wonder whether $\mathsf{plus}(x, y)$, so defined, is the addition function we know and love. What follows, given primitive recursive functions $\mathsf{idnt}^1_1(x)$ and $\mathsf{suc}(\mathsf{idnt}^3_3(x, y, u))$ is that a primitive recursive function results by recursion from them. It turns out that this is the addition function. It is left as an exercise to exhibit $\mathsf{times}(x, y)$, $\mathsf{fact}(x)$ and $\mathsf{power}(x, y)$ as primitive recursive as well.

*E12.1. (a) Show that the proposed gfact and $\mathsf{hfact}(y, u)$ result in conditions (e) and (f). Then (b) produce a defininition for $\mathsf{power}(x, y)$ by finding functions $\mathsf{gpower}(x)$, and $\mathsf{hpower}(x, y, u)$ and then show that they have the same result as conditions (g) and (h).

E12.2. Generate a sequence of functions sufficient to show that $\mathsf{power}(x, y)$ is primitive recursive.

E12.3. Install some convenient version of Ruby on your computing platform (see `http://www.ruby-lang.org/en/downloads/`). Then open `recursive1.rb` from the course website. Extend the sequence of functions started there to include `fact(x)` and `power(x,y)`. Calculate some values of these functions and print the results, along with your program (do not worry if these latter functions run slowly for even moderate values of `x` and `y`). This assignment does not require any particular computing expertise — especially, there should be no appeal to functions except from earlier in the chain. (This exercise suggests a point, to be developed in chapter 14, that recursive functions are *computable*.)

12.2 Expressing Recursive Functions

Having identified the recursive functions, we turn now to the first of two powers to be associated with theory incompleteness. In this case, it is an *expressive* power. Say that a theory is *sound* iff its axioms are true and its proof system is sound, so that all the theorems of a sound theory are true. Then the first power is this: If a theory is sound and its interpreted language expresses all the recursive functions, then it must be negation incomplete. In this section, then, we show that $\mathscr{L}_{NT}$, on its standard interpretation, expresses the recursive functions.

12.2.1 Definition and Basic Results

For a language $\mathscr{L}$, and interpretation I, suppose that for each $\mathsf{m} \in \mathsf{U}$ there is some unique variable-free term $\overline{\mathsf{m}}$ such that in the sense of definition AI, $\mathsf{I}(\overline{\mathsf{m}}) = \mathsf{m}$, so for any variable assignment d, $\mathsf{I_d}[\overline{\mathsf{m}}] = \mathsf{m}$. The simplest way for this to happen is if there is exactly one constant assigned to each member of the universe. But the standard interpretation for number theory N also has the special feature that a variable-free term is assigned to each member of U. On this interpretation, $\emptyset$, $S\emptyset$... are terms for each object. In this case, then, for any n, we simply take as $\overline{\mathsf{n}}$, $S \ldots S\emptyset$ with n repetitions of the successor operator. So $\overline{0}$ abbreviates the term $\emptyset$, $\overline{1}$ the term $S\emptyset$, etc.

Given this, we shall say that a formula $\mathcal{R}(x)$ *expresses* a relation $\mathsf{R}(\mathsf{x})$ on interpretation I, just in case if $\mathsf{m} \in \mathsf{R}$ then $\mathsf{I}[\mathcal{R}(\overline{\mathsf{m}})] = \mathsf{T}$ and if $\mathsf{m} \notin \mathsf{R}$ then $\mathsf{I}[\sim\mathcal{R}(\overline{\mathsf{m}})] = \mathsf{T}$. So the formula is true when the individual is a member of the relation and false when it is not. To express a relation on an interpretation, a formula must "say" which individuals fall under the relation. Expressing a relation is closely related to translation. A formula $\mathcal{R}(x)$ expresses a relation $\mathsf{R}(\mathsf{x})$ when every sentence $\mathcal{R}(\overline{\mathsf{m}})$ is results a good translation of the sentence $\mathsf{m} \in \mathsf{R}$ (compare chapter 5). So there is a single intended interpretation I, and a corresponding class of good translations when $\mathcal{R}(x)$ expresses $\mathsf{R}(\mathsf{x})$ on the interpretation I. Thus, generalizing,

EXr For any language $\mathscr{L}$, interpretation I, and objects $\mathsf{m_1} \ldots \mathsf{m_n} \in \mathsf{U}$, relation $\mathsf{R}(\mathsf{x_1} \ldots \mathsf{x_n})$ is *expressed* by formula $\mathcal{R}(x_1 \ldots x_n)$ iff,

(i) If $\langle \mathsf{m_1} \ldots \mathsf{m_n} \rangle \in \mathsf{R}$ then $\mathsf{I}[\mathcal{R}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n)] = \mathsf{T}$

(ii) If $\langle \mathsf{m_1} \ldots \mathsf{m_n} \rangle \notin \mathsf{R}$ then $\mathsf{I}[\sim\mathcal{R}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n)] = \mathsf{T}$

Similarly, a one-place function $\mathsf{f}(\mathsf{x})$ has members of the sort $\langle \mathsf{x}, \mathsf{v} \rangle$ and so is really a kind of two-place relation. Thus to express a function $\mathsf{f}(\mathsf{x})$, we require a formula $\mathcal{F}(x, v)$ where if $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{f}$, then $\mathsf{I}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$. It would be natural to go on to

require that if $\langle \mathsf{m}, \mathsf{a}\rangle \notin \mathsf{f}$ then $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$. However this is not necessary once we build in another feature of functions — that they have a *unique* output for each input value. Thus we shall require,

EXf For any language $\mathscr{L}$, interpretation I, and objects $\mathsf{m}_1 \ldots \mathsf{m}_n, \mathsf{a} \in \mathsf{U}$, function $\mathsf{f}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is *expressed* by formula $\mathscr{F}(x_1 \ldots x_n, v)$ iff,

if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_n\rangle, \mathsf{a}\rangle \in \mathsf{f}$ then

(i) $\mathsf{I}[\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$

(ii) $\mathsf{I}[\forall z(\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, z) \to z = \overline{\mathsf{a}})] = \mathsf{T}$

From (i), $\mathscr{F}$ is true for $\overline{\mathsf{a}}$; from (ii) any z for for which it is true is identical to $\overline{\mathsf{a}}$.

Let us illustrate these definitions with some first applications. First, on any interpretation with the required variable-free terms, the formula $x = y$ expresses the equality relation $\textsc{eq}(\mathsf{x}, \mathsf{y})$. For if $\langle \mathsf{m}, \mathsf{n}\rangle \in \textsc{eq}$ then $\mathsf{I}[\overline{\mathsf{m}}] = \mathsf{I}[\overline{\mathsf{n}}]$ so that $\mathsf{I}[\overline{\mathsf{m}} = \overline{\mathsf{n}}] = \mathsf{T}$; and if $\langle \mathsf{m}, \mathsf{n}\rangle \notin \textsc{eq}$ then $\mathsf{I}[\overline{\mathsf{m}}] \neq \mathsf{I}[\overline{\mathsf{n}}]$ so that $\mathsf{I}[\overline{\mathsf{m}} \neq \overline{\mathsf{n}}] = \mathsf{T}$. This works because $\mathsf{I}[=]$ just is the equality relation $\textsc{eq}$. Similarly, on the standard interpretation N for number theory, $\mathsf{suc}(\mathsf{x})$ is expressed by $Sx = v$, $\mathsf{plus}(\mathsf{x}, \mathsf{y})$ by $x + y = v$, and $\mathsf{times}(\mathsf{x}, \mathsf{y})$ by $x \times y = v$. Taking just the addition case, suppose $\langle\langle \mathsf{m}, \mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{plus}$; then $\mathsf{N}[\overline{\mathsf{m}} + \overline{\mathsf{n}} = \overline{\mathsf{a}}] = \mathsf{T}$. And because addition is a function, $\mathsf{N}[\forall z((\overline{\mathsf{m}} + \overline{\mathsf{n}} = z) \to z = \overline{\mathsf{a}})] = \mathsf{T}$. Again, this works because $\mathsf{N}[+]$ just is the plus function. And similarly in the other cases. Put more generally,

T12.1. For an interpretation with the required variable-free terms assigned to members of the universe: (a) If $\mathscr{R}$ is a relation symbol and $\textsc{r}$ is a relation, and $\mathsf{I}[\mathscr{R}] = \textsc{r}(\mathsf{x}_1 \ldots \mathsf{x}_n)$, then $\textsc{r}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is expressed by $\mathscr{R}x_1 \ldots x_n$. And (b) if $\mathscr{h}$ is a function symbol and h is a function and $\mathsf{I}[\mathscr{h}] = \mathsf{h}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ then $\mathsf{h}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is expressed by $\mathscr{h}x_1 \ldots x_n = v$.

It is possible to argue semantically for these claims. However, as for translation, we take the project of demonstrating expression to be one of *providing* or supplying relevant formulas. So the theorem is immediate.

Also, as we have suggested, (i) and (ii) of condition EXf taken together are sufficient to generate a condition like EXr(ii).

T12.2. Suppose function $\mathsf{f}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is expressed by formula $\mathscr{F}(x_1 \ldots x_n, y)$; then if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$, $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$.

For simplicity, consider just a one-place function $\mathsf{f(x)}$. Suppose $\mathsf{f(x)}$ is expressed by $\mathcal{F}(x, y)$ and $\langle \mathsf{m}, \mathsf{a}\rangle \notin \mathsf{f}$. Then since f is a function, there is some b such that $\langle \mathsf{m}, \mathsf{b}\rangle \in \mathsf{f}$ for $\mathsf{a} \neq \mathsf{b}$ and so $\langle \mathsf{a}, \mathsf{b}\rangle \notin \mathsf{eq}$. Suppose $\mathsf{I}[\sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{T}$; then by TI, for some d, $\mathsf{I_d}[\sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{S}$; let h be a particular assignment of this sort; so $\mathsf{I_h}[\sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{S}$; so by SF($\sim$), $\mathsf{I_h}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{S}$.

But since $\langle \mathsf{m}, \mathsf{b}\rangle \in \mathsf{f}$ by EXf(ii), $\mathsf{I}[\forall z(\mathcal{F}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{b}})] = \mathsf{T}$; so by TI, for any d, $\mathsf{I_d}[\forall z(\mathcal{F}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{b}})] = \mathsf{S}$; so $\mathsf{I_h}[\forall z(\mathcal{F}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{b}})] = \mathsf{S}$; so by SF($\forall$), $\mathsf{I}_{\mathsf{h}(z|\mathsf{a})}[\mathcal{F}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{b}}] = \mathsf{S}$; so since $\mathsf{I_h}[\overline{\mathsf{a}}] = \mathsf{a}$, by T10.2, $\mathsf{I_h}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \rightarrow \overline{\mathsf{a}} = \overline{\mathsf{b}}] = \mathsf{S}$; so by SF($\rightarrow$), $\mathsf{I_h}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{S}$ or $\mathsf{I_h}[\overline{\mathsf{a}} = \overline{\mathsf{b}}] = \mathsf{S}$; so $\mathsf{I_h}[\overline{\mathsf{a}} = \overline{\mathsf{b}}] = \mathsf{S}$; but $\mathsf{I_h}[\overline{\mathsf{a}}] = \mathsf{a}$ and $\mathsf{I_h}[\overline{\mathsf{b}}] = \mathsf{b}$; so by SF(r), $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathsf{I}[=]$; so $\langle \mathsf{a}, \mathsf{b}\rangle \in \mathsf{eq}$. This is impossible; reject the assumption: If $\mathsf{f(x)}$ is expressed by $\mathcal{F}(x, y)$ and $\langle \mathsf{m}, \mathsf{a}\rangle \notin \mathsf{f}$, then $\mathsf{I}[\sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$.

So if both $\langle \mathsf{m}, \mathsf{a}\rangle \notin \mathsf{f}$ and $\mathsf{I}[\sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{T}$, with condition EXf(i), we end up with an assignment where both $\mathsf{I_h}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{S}$ and $\mathsf{I_h}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{b}})] = \mathsf{S}$. But this violates the uniqueness constraint EXf(ii). So if $\langle \mathsf{m}, \mathsf{a}\rangle \notin \mathsf{f}$ then $\mathsf{I}[\sim\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$. So this gives us the same kind of constraint for functions as for relations.

E12.4. Provide semantic arguments to prove both parts of T12.1. So, for the first part assume that $\mathsf{I}[\mathcal{R}(x_1 \dots x_n)] = \mathsf{R}(\mathsf{x_1} \dots \mathsf{x_n})$. Then show (i) if $\langle \mathsf{m_1} \dots \mathsf{m_n}\rangle \in \mathsf{R}$ then $\mathsf{I}[\mathcal{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)] = \mathsf{T}$; and (ii) if $\langle \mathsf{m_1} \dots \mathsf{m_n}\rangle \notin \mathsf{R}$ then $\mathsf{I}[\sim\mathcal{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)] = \mathsf{T}$. And similarly for the second part based on EXf, where you may treat $\langle\langle \mathsf{m_1} \dots \mathsf{m_n}\rangle, \mathsf{a}\rangle$ as the same object as $\langle \mathsf{m_1} \dots \mathsf{m_n}, \mathsf{a}\rangle$.

12.2.2 Core Result

So far, on interpretation N, we have been able to express the relation eq, and the functions, suc, plus, and times. But our aim is to show that, on the standard interpretation N of $\mathcal{L}_{NT}$, every recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some formula $\mathcal{F}(\vec{x}, v)$.

But it is not obvious that this can be done. At least some functions must remain inexpressible in any language that has a countable vocabulary, and so in $\mathcal{L}_{NT}$. We shall see a concrete example later in the chapter. For now, consider a straightforward diagonal argument. By reasoning as from T10.7 (p. 475) there is an enumeration of all the formulas in a countable language. Isolate just formulas $\mathcal{P}_0$, $\mathcal{P}_1$, $\mathcal{P}_2$... that express functions of one variable, and consider the functions $\mathsf{f_0(x)}$, $\mathsf{f_1(x)}$, $\mathsf{f_2(x)}$... so expressed. These are all the expressible functions of one variable. Consider a grid with the functions listed down the left-hand column, and their values for each integer from left-to-right.

	0	1	2	...
$\mathsf{f}_0(\mathsf{x})$	$\mathbf{f_0(0)}$	$\mathsf{f}_0(1)$	$\mathsf{f}_0(2)$	
$\mathsf{f}_1(\mathsf{x})$	$\mathsf{f}_1(0)$	$\mathbf{f_1(1)}$	$\mathsf{f}_1(2)$	
$\mathsf{f}_2(\mathsf{x})$	$\mathsf{f}_2(0)$	$\mathsf{f}_2(1)$	$\mathbf{f_2(2)}$	
$\vdots$				

Moving along the diagonal, consider a function $\mathsf{f_d(x)}$ such that for any n, $\mathsf{f_d(n)} = \mathsf{f_n(n)} + 1$. So $\mathsf{f_d(x)}$ is, $\{\langle 0, \mathsf{f_0}(0) + 1\rangle, \langle 1, \mathsf{f_1}(1) + 1\rangle, \langle 2, \mathsf{f_2}(2) + 1\rangle, \ldots\}$. So for any integer n, this function finds the value of $\mathsf{f_n}$ along the diagonal, and adds one. But $\mathsf{f_d(x)}$ cannot be any of the expressible functions. It differs from $\mathsf{f_0(x)}$ insofar as $\mathsf{f_d}(0) \neq \mathsf{f_0}(0)$; it differs from $\mathsf{f_1(x)}$ insofar as $\mathsf{f_d}(1) \neq \mathsf{f_1}(1)$; and so forth. So $\mathsf{f_d(x)}$ is an inexpressible function. Though it has a unique output for every input value, there is no finite formula sufficient to express it.

We have already seen that $\mathsf{plus(x, y)}$ and $\mathsf{times(x, y)}$ are expresible in $\mathscr{L}_{NT}$. But there is no obvious mechanism in $\mathscr{L}_{NT}$ to express, say, $\mathsf{fact(x)}$. Given that not all functions are expressible, it is a significant matter, then, to see that all the recursive functions are expressible with interpretation N in $\mathscr{L}_{NT}$. Our main argument shall be an induction on the sequence of recursive functions. For one key case, we defer discussion into the next section.

T12.3. On the standard interpretation N of $\mathscr{L}_{NT}$, each recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some formula $\mathscr{F}(\vec{x}, v)$.

For any recursive function $\mathsf{f_a}$ there is a sequence of functions $\mathsf{f_0}$, $\mathsf{f_1} \ldots \mathsf{f_a}$ such that each member is an initial function or arises from previous members by composition, recursion or regular minimization. By induction on functions in this sequence.

Basis: $\mathsf{f_0}$ is an initial function $\mathsf{suc(x)}$, $\mathsf{zero(x)}$, or $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \ldots \mathsf{x_j})$.

(s) $\mathsf{f_0}$ is $\mathsf{suc(x)}$. Then by T12.1, $\mathsf{f_0}$ is expressed by $\mathscr{F}(x, v) =_{\text{def}} Sx = v$.

(z) $\mathsf{f_0}$ is $\mathsf{zero(x)}$. Then $\mathsf{f_0}$ is expressed by $\mathscr{F}(x, v) =_{\text{def}} x = x \wedge v = \emptyset$. Suppose $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{zero}$. Then since a is zero, $\mathsf{N}[\overline{\mathsf{m}} = \overline{\mathsf{m}} \wedge \overline{\mathsf{a}} = \emptyset] = \mathsf{T}$. And any z that is zero is equal to a — so that $\mathsf{N}[\forall z(\overline{\mathsf{m}} = \overline{\mathsf{m}} \wedge z = \emptyset \rightarrow z = \overline{\mathsf{a}})] = \mathsf{T}$.

(i) $\mathsf{f_0}$ is $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \ldots \mathsf{x_j})$. Then $\mathsf{f_0}$ is expressed by $\mathscr{F}(x_1 \ldots x_j, v) =_{\text{def}} (x_1 = x_1 \wedge \ldots \wedge x_j = x_j) \wedge x_k = v$.[4] Suppose $\langle\langle \mathsf{m}_1 \ldots \mathsf{m_j}\rangle, \mathsf{a}\rangle \in \mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}$.

[4]Perhaps it will have occurred to the reader that $\mathsf{idnt}^3_2(\mathsf{x, y, z})$, say, is expressed by $x = x \wedge z = z \wedge y = v$ as well as $x = x \wedge y = y \wedge z = z \wedge y = v$ — where the first is relatively "efficient" insofar

Then since $\mathsf{a} = \mathsf{m_k}$, $\mathsf{N}[(\overline{\mathsf{m}}_1 = \overline{\mathsf{m}}_1 \wedge \ldots \wedge \overline{\mathsf{m}}_\mathsf{j} = \overline{\mathsf{m}}_\mathsf{j}) \wedge \overline{\mathsf{m}}_\mathsf{k} = \overline{\mathsf{a}}] = \mathsf{T}$. And any $z = \mathsf{m_k}$ is equal to a — so that $\mathsf{N}[\forall z((\overline{\mathsf{m}}_1 = \overline{\mathsf{m}}_1 \wedge \ldots \wedge \overline{\mathsf{m}}_\mathsf{j} = \overline{\mathsf{m}}_\mathsf{j} \wedge \overline{\mathsf{m}}_\mathsf{k} = z) \rightarrow z = \overline{\mathsf{a}})] = \mathsf{T}$.

Assp: For any i, $0 \leq i < k$, $\mathsf{f_i}(\vec{\mathsf{x}})$ is expressed by some $\mathscr{F}(\vec{x}, v)$

Show: $\mathsf{f_k}(\mathsf{x})$ is expressed by some $\mathscr{F}(\vec{x}, v)$.

$\mathsf{f_k}$ is either an initial function or arises from previous members by composition, recursion or regular minimization. If it is an initial function then as in the basis. So suppose $\mathsf{f_k}$ arises from previous members.

(c) $\mathsf{f_k}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ arises by composition from $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$. By assumption $\mathsf{g}(\vec{\mathsf{y}})$ is expressed by some $\mathscr{G}(\vec{y}, w)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ by $\mathscr{H}(\vec{x}, w, \vec{z}, v)$; then their composition $\mathsf{f}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ is expressed by $\mathscr{F}(\vec{x}, \vec{y}, \vec{z}, v) =_{\text{def}} \exists w[\mathscr{G}(\vec{y}, w) \wedge \mathscr{H}(\vec{x}, w, \vec{z}, v)]$. For simplicity, consider a case where $\vec{\mathsf{x}}$ and $\vec{\mathsf{z}}$ drop out and $\vec{\mathsf{y}}$ is a single variable y; so $\mathscr{F}(y, v) =_{\text{def}} \exists w[\mathscr{G}(y, w) \wedge \mathscr{H}(w, v)]$. Suppose $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{f_k}$; then by composition there is some b such that $\langle \mathsf{m}, \mathsf{b} \rangle \in \mathsf{g}$ and $\langle \mathsf{b}, \mathsf{a} \rangle \in \mathsf{h}$. Because $\mathscr{G}$ and $\mathscr{H}$ express g and h, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{b}})] = \mathsf{T}$ and $\mathsf{N}[\mathscr{H}(\overline{\mathsf{b}}, \overline{\mathsf{a}})] = \mathsf{T}$; so $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{b}}) \wedge \mathscr{H}(\overline{\mathsf{b}}, \overline{\mathsf{a}})] = \mathsf{T}$, and $\mathsf{N}[\exists w(\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, \overline{\mathsf{a}}))] = \mathsf{T}$. Further, by expression, $\mathsf{N}[\forall z(\mathscr{G}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{b}})] = \mathsf{T}$ and $\mathsf{N}[\forall z(\mathscr{H}(\overline{\mathsf{b}}, z) \rightarrow z = \overline{\mathsf{a}})] = \mathsf{T}$; so that for a given m, there is just one $w = \mathsf{b}$ and so one $z = \mathsf{a}$ to satisfy the expression and $\mathsf{N}[\forall z(\exists w(\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, z)) \rightarrow z = \overline{\mathsf{a}})] = \mathsf{T}$.

(r) $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. By assumption $\mathsf{g}(\vec{\mathsf{x}})$ is expressed by some $\mathscr{G}(\vec{x}, v)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ is expressed by $\mathscr{H}(\vec{x}, y, u, v)$. And $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ is therefore expressed by means of Gödel's β-function as discussed in the next section.

(m) $\mathsf{f_k}(\vec{\mathsf{x}})$ arises by regular minimization from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$. By assumption, $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$ is expressed by some $\mathscr{G}(\vec{x}, y, z)$. Then $\mathsf{f_k}(\vec{\mathsf{x}})$ is expressed by $\mathscr{F}(\vec{x}, v) =_{\text{def}} \mathscr{G}(\vec{x}, v, \emptyset) \wedge (\forall y < v){\sim}\mathscr{G}(\vec{x}, y, \emptyset)$. Suppose $\vec{x}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{f}$; then $\langle\langle \mathsf{m}, \mathsf{a} \rangle, \mathsf{0} \rangle \in \mathsf{g}$ and for any $\mathsf{n} < \mathsf{a}$, $\langle\langle \mathsf{m}, \mathsf{n} \rangle, \mathsf{0} \rangle \notin \mathsf{g}$. So because $\mathscr{G}$ expresses g, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, \emptyset) \wedge (\forall y < \overline{\mathsf{a}}){\sim}\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)] = \mathsf{T}$. And the result is unique: for any $\mathsf{k} < \mathsf{a}$, $\mathsf{N}[\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{k}}, \emptyset)] \neq \mathsf{T}$; so the conjunction $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{k}})] \neq \mathsf{T}$. And for $\mathsf{k} > \mathsf{a}$, the other clause, $\mathsf{N}[(\forall y < \overline{\mathsf{k}}){\sim}\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)]$ fails in the case when $y = \mathsf{a}$; so the conjunction $\mathscr{F}(\overline{\mathsf{m}}, z)$ is satisfied only in the case

as it saves a conjunct. But we are after a different "efficiency" of notation and demonstration, where the formulation above serves our purposes nicely.

when z is a and $\mathsf{N}[\forall z((\mathscr{G}(\overline{\mathsf{m}}, z, \emptyset) \wedge (\forall y < z){\sim}\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)) \rightarrow z = \overline{\mathsf{a}})] = \mathsf{T}$.

Indct: Any recursive $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some $\mathscr{F}(\vec{x}, v)$

Some of the reasoning is merely sketched — however, the general idea should be clear. There might be formulas other than the stated $\mathscr{F}(\vec{x}, v)$ to express a recursive $\mathsf{f}(\vec{\mathsf{x}})$ — for example, if $\mathscr{F}(\vec{x}, v)$ expresses $\mathsf{f}(\vec{\mathsf{x}})$, then so does $\mathscr{F}(\vec{x}, v) \wedge \mathscr{A}$ for any logical truth $\mathscr{A}$. We shall see an important alternative in the following. Let us say that $\mathscr{F}(\vec{x}, v)$ so-described is the *original* formula by which $\mathsf{f}(\vec{\mathsf{x}})$ is expressed. It remains to fill out the case for the recursion clause. This is the task of the next section.

*E12.5. By the method of our core induction, write down formulas to express the following recursive functions.

a. $\mathsf{suc(zero(x))}$

b. $\mathsf{idnt}^3_2(\mathsf{x}, \mathsf{suc(zero(x))}, \mathsf{z})$

Hint: As setup for the compositions, give each function a different output variable, where the output to one is the input to the next.

*E12.6. Fill out semantic reasoning to demonstrate that proposed (original) formulas satisfy the conditions for expression for the (z), (i), (c) and (m) clauses to T12.3 — so, for example, for (c) you will apply semantic definitions to show that $\mathsf{N}[\exists w(\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, \overline{\mathsf{a}}))] = \mathsf{T}$ and that $\mathsf{N}[\forall z(\exists w(\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, z)) \rightarrow z = \overline{\mathsf{a}})] = \mathsf{T}$. Rather than go to the unabbreviated form for the bounded quantifier in case (m) it will be fine to anticipate T12.6 to apply the (obvious) semantic clause directly.

E12.7. Say a function is *μ-recursive* just in case it satisfies the conditions for the recursive functions but without the regularity requirement for minimization. So all the recursive functions are μ-recursive, but some μ-recursive functions are not recursive. Where every recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is *total* in the sense that it returns a value for every $\vec{\mathsf{x}}$ (recall the set theory reference on p. 114), some μ-recursive functions are *partial* insofar as there may be values of $\vec{\mathsf{x}}$ for which they return no value (as occurs when minimization is applied to a $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$ that never evaluates to zero). Our argument for T12.2 simply assumes that functions are recursive and so total. In the context of partial functions, EXf

would need to be augmented with the requirement that if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$, then $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$ as a third condition. Extend the argument for T12.3 to show that on the standard interpretation N of $\mathscr{L}_{\text{NT}}$, on the extended account of expression, each μ-recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some formula $\mathscr{F}(\vec{x}, v)$.

12.2.3 The β-Function

Suppose a recursive function $\mathsf{f}(\mathsf{m}, \mathsf{n}) = \mathsf{a}$. Then for the given value of m, there is a sequence $\mathsf{k}_0, \mathsf{k}_1 \ldots \mathsf{k}_n$ with $\mathsf{k}_n = \mathsf{a}$, such that k_0 takes some initial value, and each of the other members is specially related to the one before. Thus, in the simple case of $\mathsf{plus}(\mathsf{m}, \mathsf{n})$ where $\mathsf{m} = 2$, $\mathsf{k}_0 = 2$, and each k_i is the successor of the one before. So, corresponding to $2 + 5 = 7$ is the sequence,

$$2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7$$

whose first member is set by $\mathsf{gplus}(2)$, where subsequent members result from the one before by $\mathsf{plus}(2, \mathsf{Sy}) = \mathsf{hplus}(2, \mathsf{y}, \mathsf{plus}(2, \mathsf{y}))$, whose last member is 7. And, generalizing, we shall be in a position to express recursive functions if we can express the existence of *sequences* of integers so defined. We shall be able to say $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}) = \overline{\mathsf{a}}$ if we can say “there is a sequence whose first member is $\mathsf{g}(\mathsf{m})$, with members related one to another by $\mathsf{f}(\mathsf{m}, \mathsf{Sy}) = \mathsf{h}(\mathsf{m}, \mathsf{y}, \mathsf{f}(\mathsf{m}, \mathsf{y}))$, whose n^{th} member is a.” This is a mouthful. And $\mathscr{L}_{\text{NT}}$ is not obviously equipped to do it. In, particular, $\mathscr{L}_{\text{NT}}$ has straightforward mechanisms for asserting the existence of integers — but on its face, it is not clear how to assert the existence of the arbitrary sequences which result from the recursion clause.

But Gödel shows a way out. We have already seen an instance of the general strategy we shall require in our discussion of Gödel numbering from chapter 10 (p. 475). In that case, we took a sequence of integers (keyed to vocabulary), $g_0, g_1 \ldots g_n$ and collected them into a single Gödel number $G = 2^{g_0} \times 3^{g_1} \times \ldots \times \pi_n^{g_n}$ where 2, $3 \ldots \pi_n$ are the first n primes. By the fundamental theorem of arithmetic, any number has a unique prime factorization, so the original sequence is recovered from G by factoring to find the power of 2, the power of 3 and so forth. So the single integer G represents the original sequence. And $\mathscr{L}_{\text{NT}}$ has no problem expressing the existence of a single integer! Unfortunately, however, this particular way out is unavailable to us insofar as it involves exponentiation, and the resources of $\mathscr{L}_{\text{NT}}$ so-far include only S, $+$ and $\times$.[5]

[5]Some treatments begin with a language including exponentiation precisely in order to smooth the

All the same, within the resources of $\mathscr{L}_{NT}$, by the Chinese remainder theorem (whose history reaches to ancient China), there must be *pairs* of integers sufficient to represent any sequence. Consider the *remainder* function $\mathsf{rm}(\mathsf{x}, \mathsf{y})$ which returns the remainder after x is divided by y. The *remainder* of x divided by y equals z just in case $\mathsf{z} < \mathsf{y}$ and for some w, $\mathsf{x} = (\mathsf{y} \times \mathsf{w}) + \mathsf{z}$. Then let,

$$\beta(\mathsf{p}, \mathsf{q}, \mathsf{i}) =_{\text{def}} \mathsf{rm}(\mathsf{p}, [\mathsf{q} \times \mathsf{S}(\mathsf{i}) + 1])$$

So for some fixed values of p and q the β function yields different remainders for different values of i. By the Chinese remainder theorem, for any sequence $\mathsf{k}_0, \mathsf{k}_1 \ldots \mathsf{k}_n$ there are some p and q such that for $\mathsf{i} \leq \mathsf{n}$, $\beta(\mathsf{p}, \mathsf{q}, \mathsf{i}) = \mathsf{k}_i$. So p and q together code the sequence, and the β-function returns member k_i as a function of p, q and i. Intuitively, when we divide p by $\mathsf{q} \times \mathsf{S}(\mathsf{i}) + 1$, for $\mathsf{i} \leq \mathsf{n}$, the result is a series of n remainders. The theorem tells us that *any* series $\mathsf{k}_0, \mathsf{k}_1 \ldots \mathsf{k}_n$ may be so represented (see the beta function reference).

Here is a simple example. Suppose k_0, k_1 and k_2 are $5, 2, 3$. So the last subscript in the series $\mathsf{n} = 2$. Set $\mathsf{s} = \mathsf{max}(\mathsf{n}, 5, 2, 3) = 5$; and set $\mathsf{q} = \mathsf{s}! = 120$. So $\beta(\mathsf{p}, \mathsf{q}, \mathsf{i}) = \mathsf{rm}(\mathsf{p}, 120 \times \mathsf{S}(\mathsf{i}) + 1)$. So as i increases, we are looking at,

$$\mathsf{rm}(\mathsf{p}, 121) \qquad \mathsf{rm}(\mathsf{p}, 241) \qquad \mathsf{rm}(\mathsf{p}, 361)$$

But 121, 241 and 361 so constructed must have no common factor other than 1; the remainder theorem therefore tells us that as p varies between 0 and $121 \times 241 \times 361 - 1 = 10527120$ the remainders take on every possible sequence of remainder values. But the remainders will be values up to 120, 240 and 360, which is to say, $\mathsf{q} = \mathsf{s}!$ is large enough that our simple sequence must therefore appear among the sequences of remainders. In this case, $\mathsf{p} = 5219340$ gives $\mathsf{rm}(\mathsf{p}, 121) = 5$, $\mathsf{rm}(\mathsf{p}, 241) = 3$ and $\mathsf{rm}(\mathsf{p}, 361) = 2$. There may be easier ways to generate this sequence. But there is no shortage of integers (!) so there are no worries about using large ones, and by this method Gödel gives a perfectly general way to represent the arbitrary sequence.

And we can express the β-function with the resources of $\mathscr{L}_{NT}$. Thus, for $\beta(\mathsf{p}, \mathsf{q}, \mathsf{i})$,

$$\mathscr{B}(p, q, i, v) =_{\text{def}} (\exists w \leq p)[p = (S(q \times Si) \times w) + v \wedge v < S(q \times Si)]$$

So v is the remainder after p is divided by $S(q \times Si)$. And for appropriate choice of p and q, the variable v takes on the values k_0 through k_n as i runs through the values $\emptyset$ to n.

---

exposition at this stage. But our results are all the more interesting insofar as even the relatively weak $\mathscr{L}_{NT}$ retains powers sufficient for the fatal flaw.

Arithmetic for the *Beta* Function

Say $\mathsf{rm}(c, d)$ is the remainder of c/d. For a sequence, d_0, $d_1 \ldots d_n$, let $|D|$ be the product $d_0 \times d_1 \times \ldots \times d_n$. We say d_0, $d_1 \ldots d_n$ are *relatively prime* if no two members have a common factor other than 1. Then,

I. For any relatively prime sequence $d_0, d_1 \ldots d_n$, the sequences of remainders $\mathsf{rm}(c, d_0), \mathsf{rm}(c, d_1) \ldots \mathsf{rm}(c, d_n)$ as c runs from 0 to $|D| - 1$ are all different from each other.

   Suppose otherwise. Then there are c_1 and c_2, $0 \leq c_1 < c_2 < |D|$ such that $\mathsf{rm}(c_1, d_0), \mathsf{rm}(c_1, d_1) \ldots \mathsf{rm}(c_1, d_n)$ is the same as $\mathsf{rm}(c_2, d_0)$, $\mathsf{rm}(c_2, d_1) \ldots \mathsf{rm}(c_2, d_n)$. So for each d_i, $\mathsf{rm}(c_1, d_i) = \mathsf{rm}(c_2, d_i)$; say $c_1 = ad_i + r$ and $c_2 = bd_i + r$; then since the remainders are equal, $c_2 - c_1 = bd_i - ad_i$; so each d_i divides $c_1 - c_2$ evenly. So each d_i collects a distinct set of prime factors of $c_2 - c_1$; and since $c_2 - c_1$ is divided by any product of its primes, $c_2 - c_1$ is divided by $|D|$. So $|D| \leq c_2 - c_1$. But $0 \leq c_1 < c_2 < |D|$ so $c_2 - c_1 < |D|$. Reject the assumption: The sequences of remainders as c runs from 0 to $|D| - 1$ are distinct.

II. The sequences of remainders $\mathsf{rm}(c, d_0), \mathsf{rm}(c, d_1) \ldots \mathsf{rm}(c, d_n)$ as c runs from 0 to $|D| - 1$ are all the possible sequences of remainders.

   There are d_i possible remainders a number might have when divided by d_i, $(0, 1, \ldots d_i - 1)$. But if $\mathsf{rm}(c, d_0)$ takes d_0 possible values, $\mathsf{rm}(c, d_1)$ may take its d_1 values for each value of $\mathsf{rm}(c, d_0)$; etc. So the there are $|D|$ possible sequences of remainders. But as c runs from 0 to $|D| - 1$, by (I), there are $|D|$ different sequences. So there are all the possible sequences.

III. Let s be the maximum of $n, k_0, k_1 \ldots k_n$. Then for $0 \leq i < n$, the numbers $d_i = s!(i + 1) + 1$ are each greater than any k_j and are relatively prime.

   Since s is the the maximum of $n, k_0, k_1 \ldots k_n$, the first is obvious. To see that the d_i are relatively prime, suppose otherwise. Then for some j, k, $1 \leq j < k \leq n + 1$, $s!j + 1$ and $s!k + 1$ have a common factor p. But any number up to s leaves remainder 1 when dividing $s!j + 1$; so $p > s$. And since p divides $s!j + 1$ and $s!k + 1$ it divides their difference, $s!(k - j)$; but if p divides $s!$, then it does not evenly divide $s!j + 1$; so p does not divide $s!$; so p divides $k - j$. But $1 \leq j < k \leq n + 1$; so $k - j \leq n$; so $p \leq n$; so $p \leq s$. Reject the assumption: the d_i are relatively prime.

IV. For any $k_0, k_1 \ldots k_n$, we can find a pair of numbers p, q such that for $i \leq n$, $\beta(p, q, i) = k_i$.

   With s as above, set $q = s!$, and let $\beta(p, q, i) = \mathsf{rm}(p, q(i + 1) + 1)$. By (III), for $0 \leq i \leq n$ the numbers $q_i = q(i + 1) + 1$ are relatively prime. So by (II), there are all the possible sequences of remainders as p ranges from 0 to $|D|-1$. And since by (III) each of the q_i is greater than any k_i, the sequence $k_0, k_1 \ldots k_n$ is among the possible sequences of remainders. So there is some p such that the k_i are $\mathsf{rm}(p, q(i + 1) + 1)$.

Now return to our claim that when a recursive function $\mathsf{f(m,n)} = \mathsf{a}$ there is a sequence $\mathsf{k_0, k_1 \ldots k_n}$ with $\mathsf{k_n} = \mathsf{a}$ such that $\mathsf{k_0}$ takes some initial value, and each of the other members is related to the one before according to some other recursive function. More officially, a function $\mathsf{f(\vec{x}, y)} = \mathsf{z}$ just in case there is a sequence $\mathsf{k_0, k_1 \ldots k_y}$ with,

(i) $\mathsf{k_0 = g(\vec{x})}$

(ii) if $\mathsf{i < y}$, then $\mathsf{k_{Si} = h(\vec{x}, i, k_i)}$

(iii) $\mathsf{k_y = z}$

Put in terms of the β-function, this requires, $\mathsf{f(\vec{x}, y) = z}$ just in case there are some p, q such that,

(i) $\beta\mathsf{(p, q, 0) = g(\vec{x})}$

(ii) if $\mathsf{i < y}$, then $\beta\mathsf{(p, q, Si)} = \mathsf{h(\vec{x}, i,} \beta\mathsf{(p, q, i))}$

(iii) $\beta\mathsf{(p, q, y) = z}$

By assumption, $\mathsf{g(\vec{x})}$ is expressed by some $\mathscr{G}(\vec{x}, v)$ and $\mathsf{h(\vec{x}, y, u)}$ by some $\mathscr{H}(\vec{x}, y, u, v)$. So we can express the combination of these conditions as follows. $\mathsf{f(\vec{x}, y)}$ is expressed by $\mathscr{F}(\vec{x}, y, z) =_{\text{def}}$

$$\exists p \exists q \{\exists v[\mathscr{B}(p, q, \emptyset, v) \wedge \mathscr{G}(\vec{x}, v)] \wedge$$

$$(\forall i < y) \exists u \exists v [\mathscr{B}(p, q, i, u) \wedge \mathscr{B}(p, q, Si, v) \wedge \mathscr{H}(\vec{x}, i, u, v)] \wedge$$

$$\mathscr{B}(p, q, y, z)\}$$

In the case of factorial, we have $\mathscr{G}(v) =_{\text{def}} (v = S\emptyset)$ and $\mathscr{H}(y, u, v) =_{\text{def}} (v = Sy \times u)$. So the factorial function is expressed by $\mathscr{F}(y, z) =_{\text{def}}$

$$\exists p \exists q \{\exists v[\mathscr{B}(p, q, \emptyset, v) \wedge v = S\emptyset] \wedge$$

$$(\forall i < y) \exists u \exists v [\mathscr{B}(p, q, i, u) \wedge \mathscr{B}(p, q, Si, v) \wedge v = Si \times u] \wedge$$

$$\mathscr{B}(p, q, y, z)\}$$

This expression is long — particularly if expanded to unabbreviate the β-function, but it is just right. If $\langle \mathsf{n, a} \rangle \in \mathsf{fac}$, then $\mathsf{N}[\mathscr{F}(\overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$ and the expression satisfies uniqueness as well. And similarly in the general case. So with $\mathscr{L}_{\text{NT}}$ we satisfy the recursive clause for T12.3. So its demonstration is complete, and $\mathscr{L}_{\text{NT}}$ has the resources to express any recursive function.

E12.8. Suppose k_0, k_1, k_2 and k_3 are 3, 4, 0, 2. By the method of the text, find values of p and q so that $\beta(i) = k_i$. Use your values of p and q to calculate $\beta(p, q, 0)$, $\beta(p, q, 1)$, $\beta(p, q, 2)$ and $\beta(p, q, 3)$. You will need some programmable device to search for the value of p. In Ruby, a routine along the following lines, with numerical values for `a`, `b`, `c` and `d` should suffice.

```
def loop
  p = 0
  until p % a == 3 and p % b == 4 and p % c == 0 and p % d == 2
    p = p+1
    puts "p = #{p}"
  end
  return p
end
puts "p = #{loop}"
```

In Ruby `x % y` returns the remainder of `x` divided by `y`. So, for this routine, you insert the denominators and then search (by brute force) for the value of `p` that returns the right remainders. Be prepared for it to take a while!

E12.9. Produce a formula to show that $\mathscr{L}_{NT}$ expresses the $plus$ function by the initial functions with the beta function. You need not reduce the beta form to its primitive expression!

E12.10. Say a function f_k is *simple* iff there is a series of functions f_0, $f_1 \dots f_k$ such that for any $i \leq k$,

(b) f_i is $plus(x, y)$

(r) There are $a, b < i$ such that $f_i(\vec{x}, \vec{y})$ is $plus(f_a(\vec{x}), f_b(\vec{y}))$

Show that on the standard interpretation N of $\mathscr{L}_{NT}$ each simple $f(\vec{x})$ is expressed by some formula $\mathscr{F}(\vec{x}, v)$. Except for appeal to T10.2 as appropriate, you should not depend on special theorems from the text, but show your result directly from basic definitions.

12.3 Capturing Recursive Functions

The second of the powers to be associated with theory incompleteness has to do with the theory's *proof* system. If a theory is consistent and captures recursive functions, then it is negation incomplete. In this section, we show that Q, and so any theory that includes Q, captures the recursive functions.

12.3.1 Definition and Basic Results

Where expression requires that if objects stand in a given relation, then a corresponding formula be true, capture requires that when objects stand in a relation, a corresponding formula be *provable* in the theory.

CP For any language $\mathscr{L}$, interpretation I, objects $\mathsf{m}_1 \dots \mathsf{m}_n, \mathsf{a} \in \mathsf{U}$ and theory T,

(r) Relation $\mathsf{R}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is *captured* by formula $\mathcal{R}(x_1 \dots x_n, y)$ in T just in case,

(i) If $\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle \in \mathsf{R}$ then $T \vdash \mathcal{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)$

(ii) If $\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle \notin \mathsf{R}$ then $T \vdash \sim\mathcal{R}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n)$

(f) Function $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is *captured* by formula $\mathcal{F}(x_1 \dots x_n, y)$ in T just in case,

if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{a}\rangle \in \mathsf{f}$ then

(i) $T \vdash \mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$

(ii) $T \vdash \forall z(\mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, z) \rightarrow z = \overline{\mathsf{a}})$

As a first result, and to see how these definitions work, it is easy to see that in a theory at least as strong as Q, conditions (f.i) and (f.ii) combine to yield a result like (r.ii).

T12.4. If T includes Q and function $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is captured by formula $\mathcal{F}(x_1 \dots x_n, y)$ so that conditions (f.i) and (f.ii) hold, then if $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{a}\rangle \notin \mathsf{f}$ then $T \vdash \sim\mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$.

Suppose $\mathsf{f}(\mathsf{x}_1 \dots \mathsf{x}_n)$ is captured by $\mathcal{F}(x_1 \dots x_n, y)$ and $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{a}\rangle \notin \mathsf{f}$. Then, since f is a function, there is some $\mathsf{b} \neq \mathsf{a}$ such that $\langle\langle \mathsf{m}_1 \dots \mathsf{m}_n \rangle, \mathsf{b}\rangle \in \mathsf{f}$; so by (f.i), $T \vdash \mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{b}})$; and instantiating (f.ii) to $\overline{\mathsf{a}}$, $T \vdash \mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}}) \rightarrow \overline{\mathsf{a}} = \overline{\mathsf{b}}$. But since $\mathsf{a} \neq \mathsf{b}$, and T includes Q, by T8.14, $T \vdash \overline{\mathsf{a}} \neq \overline{\mathsf{b}}$; so by MT, $T \vdash \sim\mathcal{F}(\overline{\mathsf{m}}_1 \dots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$.

Our aim is to show that recursive functions are captured in Q. In chapter 8, we showed that Q correctly decides atomic formulas of $\mathscr{L}_{NT}$. As a preliminary to showing that Q captures the recursive functions, in this section we extend that result to show that Q correctly decides a broadened range of formulas.

To understand the result to which we build in this section, we need to identify some important subclasses of formulas in $\mathscr{L}_{NT}$: the Δ_0, Σ_1 and Π_1 formulas.

Δ_0 (b) If $\mathcal{P}$ is of the form $s = t$, $s < t$ or $s \leq t$ for terms s and t, then $\mathcal{P}$ is a Δ_0 formula.

(s) If $\mathcal{P}$ and $\mathcal{Q}$ are Δ_0 formulas, then so are $\sim\mathcal{P}$, and $(\mathcal{P} \to \mathcal{Q})$.

(q) If $\mathcal{P}$ is a Δ_0 formula, then so are $(\forall x \leq t)\mathcal{P}$ and $(\forall x < t)\mathcal{P}$ where x does not appear in t.

(c) Nothing else is a Δ_0 formula.

Σ_1 A formula is *strictly* Σ_1 iff it is of the form $\exists x_1 \exists x_2 \ldots \exists x_n \mathcal{P}$ for Δ_0 $\mathcal{P}$. A formula is Σ_1 iff it is logically equivalent to a strictly Σ_1 formula.

Π_1 A formula is *strictly* Π_1 iff it is of the form $\forall x_1 \forall x_2 \ldots \forall x_n \mathcal{P}$ for Δ_0 $\mathcal{P}$. A formula is Π_1 iff it is logically equivalent to a strictly Π_1 formula.

Given the soundness and adequacy of our derivation systems, we may understand equivalence in either the semantic or syntactical sense so that $\mathcal{P}$ and $\mathcal{Q}$ are equivalent just in case $\models \mathcal{P} \leftrightarrow \mathcal{Q}$ or $\vdash \mathcal{P} \leftrightarrow \mathcal{Q}$. A Δ_0 formula is (trivially) both Σ_1 and Π_1 insofar as it is preceeded by a block of zero unbounded quantifiers. We allow the usual abbreviations and so $\wedge$, $\vee$ and $\leftrightarrow$ and bounded existential quantifiers. So, for example, $n \neq \emptyset \wedge (\exists v \leq n)(SS\emptyset \times v = n)$ is Δ_0 by a tree that works like ones we have seen many times before.

$\emptyset < n$	$SS\emptyset \times v = n$	By Δ_0(b)
	$(\exists v \leq n)(SS\emptyset \times v = n)$	By Δ_0(q)
$\emptyset < n \wedge (\exists v \leq n)(SS\emptyset \times v = n)$		By Δ_0(s)

It turns out that this formula is true just in case n is an even number other than zero. For a Δ_0 formula, all is as usual, except quantifiers are bounded. Its existential quantification,

(E) $\exists n[\emptyset < n \wedge (\exists v \leq n)(SS\emptyset \times v = n)]$

is strictly Σ_1, for it consists of an (in this case single) unbounded existential quantifier followed by a Δ_0 formula. This sentence asserts the existence of an even number other than zero. Observe that,

(F) $k = k \wedge \exists n[\emptyset < n \wedge (\exists v \leq n)(SS\emptyset \times v = n)]$

is not strictly Σ_1. For it does not have the existential quantifier attached as main operator to a Δ_0 formula. However, by standard quantifier placement rules, the unbounded existential quantifier can be pulled to the main operator position to form an equivalent strictly Σ_1 sentence. Because (F) is equivalent to a sentence that is strictly

Σ_1, it too is Σ_1. Finally, by reasoning as for QN in ND, observe that the negation of a Σ_1 formula is not Σ_1 — rather it is Π_1, and the negation of a Π_1 formula is Σ_1.

We shall show that Q correctly decides Δ_0 sentences: if $\mathcal{P}$ is Δ_0 and $\mathsf{N}[\mathcal{P}] = \mathsf{T}$ then Q $\vdash_{ND} \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ then Q $\vdash_{ND} \sim\mathcal{P}$. Further, Q *proves* true Σ_1 sentences: if $\mathcal{P}$ is Σ_1 and $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then Q $\vdash_{ND} \mathcal{P}$. Observe that where $\mathcal{P}$ is Σ_1, if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$, then $\mathsf{N}[\sim\mathcal{P}] = \mathsf{T}$ — where $\sim\mathcal{P}$ is not Σ_1 at all. So, though we show Q correctly decides Δ_0 sentences and proves true Σ_1 sentences, we will not have shown that Q proves $\sim\mathcal{P}$ when $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$ and so not have shown that Q decides all Σ_1 sentences.

We begin with some preliminary theorems to set up the main result. These are not hard, but need to be wrapped up before we can attack our intended result. First some semantic theorems that work like derived clauses to SF for inequalities and bounded quantifiers. We could not obtain these in chapter 7 because they rely on theorems from chapter 8 (and since they are not inductions, they did not belong in chapter 8). However, we introduce them now in order to make progress.

T12.5. On the standard interpretation N for $\mathcal{L}_{NT}$, (i) $\mathsf{N_d}[s \leq t] = \mathsf{S}$ iff $\mathsf{N_d}[s] \leq \mathsf{N_d}[t]$, and (ii) $\mathsf{N_d}[s < t] = \mathsf{S}$ iff $\mathsf{N_d}[s] < \mathsf{N_d}[t]$.

(i) By abv $\mathsf{N_d}[s \leq t] = \mathsf{S}$ iff $\mathsf{N_d}[\exists v(v + s = t)] = \mathsf{S}$, where v is not free in s or t; by SF($\exists$), iff there is some $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{N_{d(v|m)}}[v + s = t] = \mathsf{S}$. But $\mathsf{d}(v|\mathsf{m})[v] = \mathsf{m}$; so by TA(v), $\mathsf{N_{d(v|m)}}[v] = \mathsf{m}$; so by TA(f), $\mathsf{N_{d(v|m)}}[v + s] = \mathsf{N}[+]\langle \mathsf{m}, \mathsf{N_{d(v|m)}}[s]\rangle = \mathsf{m} + \mathsf{N_{d(v|m)}}[s]$. So by SF(r), $\mathsf{N_{d(v|m)}}[v + s = t] = \mathsf{S}$ iff $\langle \mathsf{m} + \mathsf{N_{d(v|m)}}[s], \mathsf{N_{d(v|m)}}[t]\rangle \in \mathsf{N}[=]$; iff $\mathsf{m} + \mathsf{N_{d(v|m)}}[s] = \mathsf{N_{d(v|m)}}[t]$. But since v is not free in s or t, d and $\mathsf{d}(v|\mathsf{m})$ make the same assignments to variables free in s and t; so by T8.3, $\mathsf{N_d}[s] = \mathsf{N_{d(v|m)}}[s]$ and $\mathsf{N_d}[t] = \mathsf{N_{d(v|m)}}[t]$; so $\mathsf{m} + \mathsf{N_{d(v|m)}}[s] = \mathsf{N_{d(v|m)}}[t]$ iff $\mathsf{m} + \mathsf{N_d}[s] = \mathsf{N_d}[t]$; and there exists such an m just in case $\mathsf{N_d}[s] \leq \mathsf{N_d}[t]$. So $\mathsf{N_d}[s \leq t] = \mathsf{S}$ iff $\mathsf{N_d}[s] \leq \mathsf{N_d}[t]$.

(ii) is homework.

As an immediate corollary, $\mathsf{N_d}[s \leq t] \neq \mathsf{S}$ just in case $\mathsf{N_d}[s] > \mathsf{N_d}[t]$; and similarly for $>$.

T12.6. On the standard interpretation N for $\mathcal{L}_{NT}$, (i) $\mathsf{N_d}[(\forall x \leq t)\mathcal{P}] = \mathsf{S}$ iff for every $\mathsf{m} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$ and (ii), $\mathsf{N_d}[(\forall x < t)\mathcal{P}] = \mathsf{S}$ iff for every $\mathsf{m} < \mathsf{N_d}[t]$, $\mathsf{N_{d(x|m)}}[\mathcal{P}] = \mathsf{S}$.

(i) By abv $\mathsf{N_d}[(\forall x \leq t)\mathcal{P}] = \mathsf{S}$ iff $\mathsf{N_d}[\forall x(x \leq t \to \mathcal{P})] = \mathsf{S}$ where x does not appear in t; by SF($\forall$), iff for any $\mathsf{m} \in \mathsf{U}$, $\mathsf{N_{d(x|m)}}[x \leq t \to \mathcal{P}] = \mathsf{S}$; by

SF($\rightarrow$), iff for any $\mathsf{m} \in \mathsf{U}$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[x \leq t] \neq \mathsf{S}$ or $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{P}] = \mathsf{S}$; which is to say, iff for any $\mathsf{m} \in \mathsf{U}$, if $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[x \leq t] = \mathsf{S}$, then $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{P}] = \mathsf{S}$. But $\mathsf{d}(x|\mathsf{m})[x] = \mathsf{m}$; so $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[x] = \mathsf{m}$; and since x is not free in t, d and $\mathsf{d}(x|\mathsf{m})$ agree on assignments to variables free in t; so by T8.3, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[t] = \mathsf{N}_{\mathsf{d}}[t]$; so with T12.5, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[x \leq t] = \mathsf{S}$ iff $\mathsf{m} \leq \mathsf{N}_{\mathsf{d}}[t]$; so $\mathsf{N}_{\mathsf{d}}[(\forall x \leq t)\mathcal{P}] = \mathsf{S}$ iff for any m, if $\mathsf{m} \leq \mathsf{N}_{\mathsf{d}}[t]$, then $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{P}] = \mathsf{S}$.

(ii) is homework.

T12.7. On the standard interpretation N for $\mathcal{L}_{NT}$, (i) $\mathsf{N}_{\mathsf{d}}[(\exists x \leq t)\mathcal{P}] = \mathsf{S}$ iff for some $\mathsf{m} \leq \mathsf{N}_{\mathsf{d}}[t]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{P}] = \mathsf{S}$ and (ii), $\mathsf{N}_{\mathsf{d}}[(\exists x < t)\mathcal{P}] = \mathsf{S}$ iff for some $\mathsf{m} < \mathsf{N}_{\mathsf{d}}[t]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathcal{P}] = \mathsf{S}$.

Homework

We are finally ready for the results to which we have been building: First, Q correctly decides Δ_0 sentences of $\mathcal{L}_{NT}$.

T12.8. For any Δ_0 sentence $\mathcal{P}$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathsf{Q} \vdash_{ND} \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$, then $\mathsf{Q} \vdash_{ND} \sim\mathcal{P}$.

By induction on the number of operators in $\mathcal{P}$.

Basis: If $\mathcal{P}$ is an an atomic Δ_0 sentence it is $t = s$, $t \leq s$ or $t < s$. So by T8.14, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, $\mathsf{Q} \vdash_{ND} \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$, $\mathsf{Q} \vdash_{ND} \sim\mathcal{P}$.

Assp: For any $i, 0 \leq i < k$, if a Δ_0 setntence $\mathcal{P}$ has i operator symbols, then if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, $\mathsf{Q} \vdash_{ND} \mathcal{P}$ and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$, $\mathsf{Q} \vdash_{ND} \sim\mathcal{P}$.

Show: If a Δ_0 sentence $\mathcal{P}$ has k operator symbols, then if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, $\mathsf{Q} \vdash_{ND} \mathcal{P}$ and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$, $\mathsf{Q} \vdash_{ND} \sim\mathcal{P}$.

If a Δ_0 sentence $\mathcal{P}$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{A} \rightarrow \mathcal{B}$, $(\forall x \leq t)\mathcal{A}$ or $(\forall x < t)\mathcal{A}$ where $\mathcal{A}$, $\mathcal{B}$ have $< k$ operator symbols and x does not appear in t.

($\sim$) $\mathcal{P}$ is $\sim\mathcal{A}$. (i) Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[\sim\mathcal{A}] = \mathsf{T}$; so by T8.6, $\mathsf{N}[\mathcal{A}] \neq \mathsf{T}$; so by assumption, $\mathsf{Q} \vdash_{ND} \sim\mathcal{A}$; so $\mathsf{Q} \vdash_{ND} \mathcal{P}$. (ii) Suppose $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$; then $\mathsf{N}[\sim\mathcal{A}] \neq \mathsf{T}$; so by T8.6, $\mathsf{N}[\mathcal{A}] = \mathsf{T}$; so by assumption $\mathsf{Q} \vdash_{ND} \mathcal{A}$; so by DN, $\mathsf{Q} \vdash_{ND} \sim\sim\mathcal{A}$; so $\mathsf{Q} \vdash_{ND} \sim\mathcal{P}$.

($\rightarrow$) $\mathcal{P}$ is $\mathcal{A} \rightarrow \mathcal{B}$. (i) Suppose $\mathsf{N}[\mathcal{A} \rightarrow \mathcal{B}] = \mathsf{T}$; then by T8.6, $\mathsf{N}[\mathcal{A}] \neq \mathsf{T}$ or $\mathsf{N}[\mathcal{B}] = \mathsf{T}$. So by assumption, $\mathsf{Q} \vdash_{ND} \sim\mathcal{A}$ or $\mathsf{Q} \vdash_{ND} \mathcal{B}$. So by $\vee$I twice $\mathsf{Q} \vdash_{ND} \sim\mathcal{A} \vee \mathcal{B}$ or $\mathsf{Q} \vdash_{ND} \sim\mathcal{A} \vee \mathcal{B}$; so $\mathsf{Q} \vdash_{ND} \sim\mathcal{A} \vee \mathcal{B}$; so by Impl, $\mathsf{Q} \vdash_{ND} \mathcal{A} \rightarrow \mathcal{B}$. Part (ii) is homework.

($\forall \leq$) $\mathcal{P}$ is $(\forall x \leq t)\mathcal{A}(x)$. Since $\mathcal{P}$ is a sentence, x is the only variable free in $\mathcal{A}$; in particular, since x does not appear in t, t must be variable-free; so $\mathsf{N_d}[t] = \mathsf{N}[t]$ and where $\mathsf{N}[t] = \mathsf{n}$, by T8.13, $\mathsf{Q} \vdash_{ND} t = \bar{\mathsf{n}}$; so by =E, $\mathsf{Q} \vdash_{ND} \mathcal{P}$ just in case $\mathsf{Q} \vdash_{ND} (\forall x \leq \bar{\mathsf{n}})\mathcal{A}(x)$.

(i) Suppose $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then $\mathsf{N}[(\forall x \leq t)\mathcal{A}(x)] = \mathsf{T}$; so by TI, for any d, $\mathsf{N_d}[(\forall x \leq t)\mathcal{A}(x)] = \mathsf{S}$; so by T12.6, for any $\mathsf{m} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] = \mathsf{S}$; so where $\mathsf{N_d}[\mathsf{t}] = \mathsf{N}[t] = \mathsf{n}$, for any $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] = \mathsf{S}$; but $\mathsf{N_d}[\bar{\mathsf{m}}] = \mathsf{m}$, so with T10.2, for any $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N_d}[\mathcal{A}(\bar{\mathsf{m}})] = \mathsf{S}$; since x is the only variable free in $\mathcal{A}$, $\mathcal{A}(\bar{\mathsf{m}})$ is a sentence; so with T8.5, for any $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N}[\mathcal{A}(\bar{\mathsf{m}})] = \mathsf{T}$; so $\mathsf{N}[\mathcal{A}(\emptyset)] = \mathsf{T}$ and $\mathsf{N}[\mathcal{A}(\bar{1})] = \mathsf{T}$ and ... and $\mathsf{N}[\mathcal{A}(\bar{\mathsf{n}})] = \mathsf{T}$; so by assumption, $\mathsf{Q} \vdash_{ND} \mathcal{A}(\emptyset)$ and $\mathsf{Q} \vdash_{ND} \mathcal{A}(\bar{1})$ and ... and $\mathsf{Q} \vdash_{ND} \mathcal{A}(\bar{\mathsf{n}})$; so by T8.21, $\mathsf{Q} \vdash_{ND} (\forall x \leq \bar{\mathsf{n}})\mathcal{A}(x)$; so with our preliminary result, $\mathsf{Q} \vdash_{ND} \mathcal{P}$.

(ii) Suppose $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$; then $\mathsf{N}[(\forall x \leq t)\mathcal{A}(x)] \neq \mathsf{T}$; so by TI, for some d, $\mathsf{N_d}[(\forall x \leq t)\mathcal{A}(x)] \neq \mathsf{S}$; so by T12.6, for some $\mathsf{m} \leq \mathsf{N_d}[t]$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] \neq \mathsf{S}$; so where $\mathsf{N_d}[\mathsf{t}] = \mathsf{N}[t] = \mathsf{n}$, for some $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N_{d(x|m)}}[\mathcal{A}(x)] \neq \mathsf{S}$; but $\mathsf{N_d}[\bar{\mathsf{m}}] = \mathsf{m}$, so with T10.2, for some $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N_d}[\mathcal{A}(\bar{\mathsf{m}})] \neq \mathsf{S}$; so by TI, for some $\mathsf{m} \leq \mathsf{n}$, $\mathsf{N}[\mathcal{A}(\bar{\mathsf{m}})] \neq \mathsf{T}$; so by assumption for some $\mathsf{m} \leq \mathsf{n}$, $\mathsf{Q} \vdash_{ND} {\sim}\mathcal{A}(\bar{\mathsf{m}})$; so by T8.20, $\mathsf{Q} \vdash_{ND} (\exists x \leq \bar{\mathsf{n}}){\sim}\mathcal{A}(x)$; so by bounded quantifier negation (BQN), $\mathsf{Q} \vdash_{ND} {\sim}(\forall x \leq \bar{\mathsf{n}})\mathcal{A}(x)$; so with our preliminary result, $\mathsf{Q} \vdash_{ND} {\sim}\mathcal{P}$.

($\forall <$) homework.

Indct: So for any Δ_0 sentence $\mathcal{P}$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathsf{Q} \vdash_{ND} \mathcal{P}$, and if $\mathsf{N}[\mathcal{P}] \neq \mathsf{T}$, then $\mathsf{Q} \vdash_{ND} {\sim}\mathcal{P}$.

And now, Q proves true Σ_1 sentences.

T12.9. For any (strict) Σ_1 sentence $\mathcal{P}$ if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathsf{Q} \vdash_{ND} \mathcal{P}$.

This is a simple induction on the number of unbounded existential quantifiers in $\mathcal{P}$. Hint: If $\mathcal{P}$ has no unbounded existential quantifiers, then it is Δ_0. Otherwise, if $\exists x\mathcal{P}$ is true, it will be easy to show that for some m, $\mathcal{P}(\bar{\mathsf{m}})$ is true; you can then apply your assumption, and $\exists$I.

Corollary: For any Σ_1 sentence $\mathcal{P}$, if $\mathsf{N}[\mathcal{P}] = \mathsf{T}$, then $\mathsf{Q} \vdash_{ND} \mathcal{P}$. Suppose a Σ_1 $\mathcal{P}$ is such that $\mathsf{N}[\mathcal{P}] = \mathsf{T}$; then by equivalence there is some strict Σ_1 $\mathcal{P}^*$ such that $\mathsf{N}[\mathcal{P}^*] = \mathsf{T}$; so by the main theorem, $\mathsf{Q} \vdash_{ND} \mathcal{P}^*$; and by equivalence again, $\mathsf{Q} \vdash_{ND} \mathcal{P}$.

This completes what we set out to show in this subsection. These results should seem intuitive: Q proves results about particular numbers, $1 + 1 = 2$ and the like. But Δ_0 sentences assert (potentially complex) particular facts about numbers — and we show that Q proves any Δ_0 sentence. Similarly, any Σ_1 sentence is true *because* of some particular fact about numbers; since Q proves that particular fact, it is sufficient to prove the Σ_1 sentence.

E12.11. Complete the demonstration of T12.5 - T12.7 by showing the remaining parts. These should be straightforward, given parts worked in the text.

*E12.12. (i) Complete the demonstration of T12.8 by finishing the remaining cases. You should set up the entire argument, but may appeal to the text for parts already completed, as the text appeals to homework. (ii) Show directly cases $(\exists \leq)$ and $(\exists <)$.

E12.13. Provide an argument to demonstrate T12.9.

12.3.2 Basic Result

We now set out to show that Q captures all the recursive functions. We begin showing that the original formulas by which we have expressed recursive functions are Σ_1. After that, we get our result in in two forms. First a straightforward basic version. However, this version gets a result slightly weaker than the one we would like. But it is easily strengthened to the final form.

First, then, an argument that the original formulas by which we have expressed recursive functions are Σ_1. This argument merely reviews the strategy from T12.3 for expression to show that each formula is equivalent to a strictly Σ_1 formula and so is Σ_1.

T12.10. The original formula by which any recursive function is expressed is Σ_1.

By induction on the sequence of recursive functions.

Basis: From T12.3, $\mathsf{suc}(\mathsf{x})$ is originally expressed by $Sx = v$; $\mathsf{zero}(\mathsf{x})$ by $x = x \wedge v = \emptyset$ and $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \ldots \mathsf{x}_{\mathsf{j}})$ by $(x_1 = x_1 \wedge \ldots \wedge x_j = x_j) \wedge x_k = v$. These are all Δ_0, and therefore Σ_1.

Assp: For any any i, $0 \leq i < k$, the original formula $\mathscr{F}(\vec{x}, v)$ by which $\mathsf{f_i}(\vec{\mathsf{x}})$ is expressed is Σ_1

Show: The original formula $\mathscr{F}(\vec{x}, v)$ by which $\mathsf{f_k}(\vec{\mathsf{x}})$ is expressed is Σ_1

$\mathsf{f_k}$ is either an initial function or arises from previous members by composition, recursion or regular minimization. If it is an initial function, then as in the basis. So suppose $\mathsf{f_k}$ arises from previous members.

(c) $\mathsf{f_k}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ arises by composition from $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$. By assumption $\mathsf{g}(\vec{\mathsf{y}})$ is expressed by some Σ_1 formula equivalent to $\exists\vec{j}\mathscr{G}(\vec{y}, w)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ by a Σ_1 formula equivalent to $\exists\vec{k}\mathscr{H}(\vec{x}, w, \vec{z}, v)$ where $\mathscr{G}$ and $\mathscr{H}$ are individually Δ_0. Then their original composition $\mathscr{F}(\vec{x}, \vec{y}, \vec{z}, v)$ is equivalent to $\exists w[\exists\vec{j}\mathscr{G}(\vec{y}, w) \wedge \exists\vec{k}\mathscr{H}(\vec{x}, w, \vec{z}, v)]$; and by standard quantifier placement rules, this is equivalent to $\exists w\exists\vec{j}\exists\vec{k}[\mathscr{G}(\vec{y}, w) \wedge \mathscr{H}(\vec{x}, w, \vec{z}, v)]$, where this is Σ_1.

(r) $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. By assumption $\mathsf{g}(\vec{\mathsf{x}})$ is expressed by some Σ_1 formula $\exists\vec{j}\mathscr{G}(\vec{x}, v)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by $\exists\vec{k}\mathscr{H}(\vec{x}, y, u, v)$. And, as before, the β-function $\mathscr{B}(p, q, i, v)$ is expressed by,

$$(\exists w \leq p)[p = (S(q \times Si) \times w) + v \wedge v < S(q \times Si)]$$

where this is Δ_0. Then the original formula $\mathscr{F}(\vec{x}, y, z)$ by which $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ is expressed is equivalent to,

$\exists p\exists q\{\exists v[\mathscr{B}(p, q, \emptyset, v) \wedge \exists\vec{j}\mathscr{G}(\vec{x}, v)] \wedge$

$(\forall i < y)\exists u\exists v[\mathscr{B}(p, q, i, u) \wedge \mathscr{B}(p, q, Si, v) \wedge \exists\vec{k}\mathscr{H}(\vec{x}, i, u, v)] \wedge \mathscr{B}(p, q, y, z)\}$

This time, standard quantifier placement rules are not enough to identify the formula as Σ_1. We can pull the initial v and $\vec{j}$ quantifiers out. And the $\vec{k}$ quantifiers come out with the u and v quantifiers. The problem is getting these past the bounded universal i quantifier.

For this, we use a sort of trick: For a simplified case, consider $(\forall i < y)\exists v\mathscr{P}(i, v)$; this requires that for each $\mathsf{i} < \mathsf{y}$ there is at least one v that makes $\mathscr{P}(i, v)$ true; for each $\mathsf{i} < \mathsf{y}$ consider the least such v, and let a be the greatest member of this collection. Then $(\forall i < y)(\exists v < \overline{\mathsf{a}})\mathscr{P}(i, v)$ says the same as the original expression. And therefore, no matter what y may be, $\exists j(\forall i < y)(\exists v < j)\mathscr{P}(i, v)$ is true iff the original expression is true. So the existential quantifier comes past the bounded universal, leaving behind a bounded existential "shadow." Thus the existential u, v and $\vec{k}$ quantifiers come to the front, and the result is Σ_1.

(m) $\mathsf{f_k}(\vec{\mathsf{x}})$ arises by regular minimization from $\mathsf{g}(\vec{\mathsf{x}},\mathsf{y})$. By assumption, $\mathsf{g}(\vec{\mathsf{x}},\mathsf{y})$ is expressed by some $\exists\vec{j}\mathscr{G}(\vec{x},y,z)$. Then the original expression by which $\mathsf{f_k}(\vec{\mathsf{x}})$ is expressed is equivalent to $\exists\vec{j}\mathscr{G}(\vec{x},v,\emptyset)\wedge(\forall y<v){\sim}\exists\vec{j}\mathscr{G}(\vec{x},y,\emptyset)$; but since $\mathscr{G}$ expresses a function, ${\sim}\exists\vec{j}\mathscr{G}(\vec{x},y,\emptyset)$ just when $\exists z[\exists\vec{j}\mathscr{G}(\vec{x},y,z)\wedge z\neq\emptyset]$; so the original expression is equivalent to, $\exists\vec{j}\mathscr{G}(\vec{x},v,\emptyset)\wedge(\forall y<v)\exists z[\exists\vec{j}\mathscr{G}(\vec{x},y,z)\wedge z\neq\emptyset]$. The first set of j quantifiers come directly to the front, and the second set, together with the z quantifier come out, as in the previous case, leaving bounded existential quantifiers behind. So the result is Σ_1.

Indct: The original formula by which any recursive function is expressed is Σ_1.

It is not proper to drag an existential quantifier out past a universal quantifier; however, it is legitimate to drag an existential past a *bounded* universal, with a bounded existential quantifier left behind as “shadow” or “witness.”

Now for our main result. Here is the sense in which our result is weaker than we might like: Rather than Q, let us suppose we are in a system $\mathrm{Q_s}$, *strengthened* Q, which has (as an axiom or) a theorem *uniqueness of remainder*,

$$\forall x\forall y[((\exists w\leq m)[m=Sn\times w+x\wedge x<Sn]\wedge(\exists w\leq m)[m=Sn\times w+y\wedge y<Sn])\rightarrow x=y]$$

for any x and y, if x is the remainder of $m/(n+1)$ and y is the remainder of $m/(n+1)$ then $x=y$. As we shall see for *Def*[*rm*] in chapter 13, PA is a system of this sort though, insofar as m and n are free variables rather than numerals, Q is not. Notice that m and n are free in this formulation; if they are instantiated to p and $q\times Si$ respectively, from uniqueness for remainder there immediately follows a parallel uniqueness result for the β-function.

$$\forall x\forall y[(\mathscr{B}(p,q,i,x)\wedge\mathscr{B}(p,q,i,y))\rightarrow x=y]$$

Further, if $\langle\langle\mathsf{p},\mathsf{q},\mathsf{i}\rangle,\mathsf{a}\rangle\in\beta$ then since $\mathscr{B}$ expresses the β-function, $\mathsf{N}[\mathscr{B}(\overline{\mathsf{p}},\overline{\mathsf{q}},\overline{\mathsf{i}},\overline{\mathsf{a}})]=\mathsf{T}$; and since $\mathscr{B}$ is Δ_0, by T12.8, Q $\vdash_{ND}\mathscr{B}(\overline{\mathsf{p}},\overline{\mathsf{q}},\overline{\mathsf{i}},\overline{\mathsf{a}})$. From this, with uniqueness, it is immediate with $\forall$E that $\mathrm{Q_s}\vdash_{ND}\forall z[\mathscr{B}(\overline{\mathsf{p}},\overline{\mathsf{q}},\overline{\mathsf{i}},z)\rightarrow z=\overline{\mathsf{a}})$. So $\mathscr{B}$ *captures* β in $\mathrm{Q_s}$.

Now we are positioned to offer a perfectly straightforward argument for capture of the recursive functions in $\mathrm{Q_s}$. Again our main argument is an induction on the sequence of recursive functions. We show that $\mathrm{Q_s}$ captures the initial functions, and then that it captures functions from composition, recursion and regular minimization.

T12.11. On the standard interpretation N for $\mathcal{L}_{\text{NT}}$, any recursive function is captured in Q_s by the original formula by which it is expressed.

By induction on the sequence of recursive functions.

Basis: $\mathsf{f_0}$ is an initial function $\mathsf{suc(x)}$, $\mathsf{zero(x)}$, or $\mathsf{idnt^j_k(x_1 \ldots x_j)}$.

(s) The original formula $\mathcal{F}(x, v)$ by which $\mathsf{suc(x)}$ is expressed is $Sx = v$. Suppose $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{suc}$.

(i) Since $Sx = v$ expresses $\mathsf{suc(x)}$, $\mathsf{N}[S\overline{\mathsf{m}} = \overline{\mathsf{a}}] = \mathsf{T}$; so, since it is Δ_0, by T12.8, Q $\vdash_{ND}$ $S\overline{\mathsf{m}} = \overline{\mathsf{a}}$; so $\text{Q}_\text{s} \vdash_{ND} \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Reason as follows,

1.	$S\overline{\mathsf{m}} = \overline{\mathsf{a}}$	from (i)
2.	$\quad S\overline{\mathsf{m}} = j$	A (g, →I)
3.	$\quad j = \overline{\mathsf{a}}$	1,2 =E
4.	$S\overline{\mathsf{m}} = j \to j = \overline{\mathsf{a}}$	2-3 →I
5.	$\forall z(S\overline{\mathsf{m}} = z \to z = \overline{\mathsf{a}})$	4 ∀I

So $\text{Q}_\text{s} \vdash_{ND} \forall z[\mathcal{F}(\overline{\mathsf{m}}, z) \to z = \overline{\mathsf{a}}]$.

(oth) It is left as homework to show that $\mathsf{zero(x)}$ is captured by $x = x \wedge v = \emptyset$ and $\mathsf{idnt^j_k(x_1 \ldots x_j)}$ by $(x_1 = x_1 \wedge \ldots \wedge x_j = x_j) \wedge x_k = v$.

Assp: For any i, $0 \le i < k$, $\mathsf{f_i(\vec{x})}$ is captured in Q_s by the original formula by which it is expressed.

Show: $\mathsf{f_k(\vec{x})}$ is captured in Q_s by the original formula by which it is expressed.

$\mathsf{f_k}$ is either an initial function or arises from previous members by composition, recursion or regular minimization. If it is an initial function, then as in the basis. So suppose $\mathsf{f_k}$ arises from previous members.

(c) $\mathsf{f_k(\vec{x}, \vec{y}, \vec{z})}$ arises by composition from $\mathsf{g(\vec{y})}$ and $\mathsf{h(\vec{x}, w, \vec{z})}$. By assumption $\mathsf{g(\vec{y})}$ is captured by some $\mathcal{G}(\vec{y}, w)$ and $\mathsf{h(\vec{x}, w, \vec{z})}$ by $\mathcal{H}(\vec{x}, w, \vec{z}, v)$; the original formula $\mathcal{F}(\vec{x}, \vec{y}, \vec{z}, v)$ by which the composition $\mathsf{f(\vec{x}, \vec{y}, \vec{z})}$ is expressed is $\exists w[\mathcal{G}(\vec{y}, w) \wedge \mathcal{H}(\vec{x}, w, \vec{z}, v)]$. For simplicity, consider a case where $\vec{\mathsf{x}}$ and $\vec{\mathsf{z}}$ drop out and $\vec{\mathsf{y}}$ is a single variable y. Suppose $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{f_k}$; then by composition there is some b such that $\langle \mathsf{m}, \mathsf{b} \rangle \in \mathsf{g}$ and $\langle \mathsf{b}, \mathsf{a} \rangle \in \mathsf{h}$.

(i) Since $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{f_k}$, and $\mathcal{F}(y, v)$ expresses f, $\mathsf{N}[\mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$; so, since $\mathcal{F}(y, v)$ is Σ_1, by T12.9, $\text{Q}_\text{s} \vdash_{ND} \mathcal{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Since $\mathcal{G}(y, w)$ captures $\mathsf{g(y)}$ and $\mathcal{H}(w, v)$ captures $\mathsf{h(w)}$, by assumption $\text{Q}_\text{s} \vdash_{ND} \forall z(\mathcal{G}(\overline{\mathsf{m}}, z) \to z = \overline{\mathsf{b}})$ and $\text{Q}_\text{s} \vdash_{ND} \forall z(\mathcal{H}(\overline{\mathsf{b}}, z) \to$

$z = \overline{\mathsf{a}}$). It is then a simple derivation for you to show that $\mathrm{Q_s} \vdash_{ND} \forall z(\exists w[\mathscr{G}(\overline{\mathsf{m}}, w) \wedge \mathscr{H}(w, z)] \to z = \overline{\mathsf{a}})$.

(r) $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. By assumption $\mathsf{g}(\vec{\mathsf{x}})$ is captured by some $\mathscr{G}(\vec{x}, v)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by $\mathscr{H}(\vec{x}, y, u, v)$; the original formula $\mathscr{F}(\vec{x}, y, z)$ by which $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ is expressed is,

$\exists p \exists q\{\exists v[\mathscr{B}(p, q, \emptyset, v) \wedge \mathscr{G}(\vec{x}, v)] \wedge (\forall i < y)\exists u \exists v[\mathscr{B}(p, q, i, u) \wedge \mathscr{B}(p, q, Si, v) \wedge \mathscr{H}(\vec{x}, i, u, v)] \wedge \mathscr{B}(p, q, y, z)\}$

Suppose $\vec{\mathsf{x}}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{n}, \mathsf{a}\rangle \in \mathsf{f_k}$. (i) Then since $\mathscr{F}(x, y, z)$ expresses f, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})] = \mathsf{T}$; so, since $\mathscr{F}(x, y, z)$ is Σ_1, by T12.9, $\mathrm{Q_s} \vdash_{ND} \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, \overline{\mathsf{a}})$. And (ii) by T12.12, immediately following, $\mathrm{Q_s} \vdash_{ND} \forall w[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \to w = \overline{\mathsf{a}}]$.

(m) $\mathsf{f_k}(\vec{\mathsf{x}})$ arises by regular minimization from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$. By assumption, $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$ is captured by some $\mathscr{G}(\vec{x}, y, z)$; the original formula by $\mathscr{F}(\vec{x}, v)$ by which $\mathsf{f_k}(\vec{\mathsf{x}})$ is expressed is $\mathscr{G}(\vec{x}, v, \emptyset) \wedge (\forall y < v)\sim\mathscr{G}(\vec{x}, y, \emptyset)$. Suppose $\vec{x}$ reduces to a single variable and $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$.

(i) Since $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f_k}$, and $\mathscr{F}(x, v)$ expresses f, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$; so since $\mathscr{F}(x, v)$ is Σ_1, by T12.9, $\mathrm{Q_s} \vdash_{ND} \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Reason as follows,

1.	$\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, \emptyset) \wedge (\forall y < \overline{\mathsf{a}})\sim\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)$	from (i)
2.	$j < \overline{\mathsf{a}} \vee j = \overline{\mathsf{a}} \vee \overline{\mathsf{a}} < j$	T8.19
3.	$\mathscr{G}(\overline{\mathsf{m}}, j, \emptyset) \wedge (\forall y < j)\sim\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)$	A (g, $\to$I)
4.	$j < \overline{\mathsf{a}}$	A (c, $\sim$I)
5.	$\mathscr{G}(\overline{\mathsf{m}}, j, \emptyset)$	3 $\wedge$E
6.	$(\forall y < \overline{\mathsf{a}})\sim\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)$	1 $\wedge$E
7.	$\sim\mathscr{G}(\overline{\mathsf{m}}, j, \emptyset)$	6,4 ($\forall$E)
8.	$\bot$	5,7 $\bot$I
9.	$j \not< \overline{\mathsf{a}}$	4-8 $\sim$I
10.	$\overline{\mathsf{a}} < j$	A (c, $\sim$I)
11.	$\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, \emptyset)$	1 $\wedge$E
12.	$(\forall y < j)\sim\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)$	3 $\wedge$E
13.	$\sim\mathscr{G}(\overline{\mathsf{m}}, \overline{\mathsf{a}}, \emptyset)$	12,10 ($\forall$E)
14.	$\bot$	11,13 $\bot$I
15.	$\overline{\mathsf{a}} \not< j$	10-14 $\sim$I
16.	$j = \overline{\mathsf{a}}$	2,9,15 DS
17.	$[\mathscr{G}(\overline{\mathsf{m}}, j, \emptyset) \wedge (\forall y < j)\sim\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)] \to j = \overline{\mathsf{a}}$	3-16 $\to$I
18.	$\forall z([\mathscr{G}(\overline{\mathsf{m}}, z, \emptyset) \wedge (\forall y < z)\sim\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)] \to z = \overline{\mathsf{a}})$	17 $\forall$I

So $\mathrm{Q_s} \vdash_{ND} \forall z([\mathscr{G}(\overline{\mathsf{m}}, z, \emptyset) \wedge (\forall y < z)\sim\mathscr{G}(\overline{\mathsf{m}}, y, \emptyset)] \to z = \overline{\mathsf{a}})$.

Indct: Any recursive $\mathsf{f}(\vec{\mathsf{x}})$ is captured by the original formula by which it is expressed in $\mathrm{Q_s}$.

For this argument, we simply rely on the ability of Q to prove particular truths, and so the Σ_1 sentences that express recursive functions. The uniqueness clauses are not Σ_1, so we have to show them directly. The case for recursion remains outstanding, and is addressed in the theorem immediately following.

T12.12. Suppose $\mathsf{f}(\vec{\mathsf{x}},\mathsf{y})$ results by recursion from functions $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{u})$ where $\mathsf{g}(\vec{\mathsf{x}})$ is captured by some $\mathscr{G}(\vec{x},v)$ and $\mathsf{h}(\vec{\mathsf{x}},\mathsf{y},\mathsf{u})$ by $\mathscr{H}(\vec{x},y,u,v)$. Then for the original expression $\mathscr{F}(\vec{x},y,z)$ of $\mathsf{f}(\vec{\mathsf{x}},\mathsf{y})$, if $\langle\langle \mathsf{m}_1\ldots\mathsf{m}_b,\mathsf{n}\rangle,\mathsf{a}\rangle\in\mathsf{f}$, $\mathrm{Q_s}\vdash\forall w[\mathscr{F}(\overline{\mathsf{m}}_1\ldots\overline{\mathsf{m}}_b,\overline{\mathsf{n}},w)\rightarrow w=\overline{\mathsf{a}}]$.

Suppose $\vec{\mathsf{x}}$ reduces to a single variable and $\langle\mathsf{m},\mathsf{n},\mathsf{a}\rangle\in\mathsf{f}$. When $\langle\mathsf{m},\mathsf{n},\mathsf{a}\rangle\in\mathsf{f}$, there are $\mathsf{k}_0\ldots\mathsf{k}_n$ such that $\mathsf{k}_n=\mathsf{a}$, $\mathsf{k}_0=\mathsf{g}(\mathsf{m})$; for $0\le\mathsf{i}<\mathsf{n}$, there are p,q such that $\beta(\mathsf{p},\mathsf{q},\mathsf{i})=\mathsf{k}_i$, $\beta(\mathsf{p},\mathsf{q},\mathsf{Si})=\mathsf{k}_{\mathsf{Si}}$, and $\mathsf{h}(\mathsf{m},\mathsf{i},\mathsf{k}_i)=\mathsf{k}_{\mathsf{Si}}$. The argument is by induction on the value of n from $\mathsf{f}(\mathsf{m},\mathsf{n})=\mathsf{a}$. Observe that $\mathscr{F}$ is long, and we shall better be able to manage the formulas given its general form $\exists p\exists q[\mathscr{A}\wedge\mathscr{C}\wedge\mathscr{B}]$. Given the structure of the definition for this recursion clause, it will be convenient to lapse into induction scheme III from the induction schemes reference on p. 384, making the assumption for a single member of the series n, and then showing that it holds for the next. Thus, assuming that $\mathrm{Q_s}\vdash\forall w[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}},w)\rightarrow w=\overline{\mathsf{k}}_n]$, we show $\mathrm{Q_s}\vdash\forall w[\mathscr{F}(\overline{\mathsf{m}},S\overline{\mathsf{n}},w)\rightarrow w=\overline{\mathsf{k}}_{\mathsf{Sn}}]$.

Basis: Suppose $\mathsf{n}=0$. From capture, $\mathrm{Q_s}\vdash_{ND}\forall z[\mathscr{G}(\overline{\mathsf{m}},z)\rightarrow z=\overline{\mathsf{k}}_0]$. By uniqueness of remainder (and generalizing on p and q), $\mathrm{Q_s}\vdash_{ND}\forall p\forall q\forall x\forall y[(\mathscr{B}(p,q,\emptyset,x)\wedge\mathscr{B}(p,q,\emptyset,y))\rightarrow x=y]$. $\mathscr{F}$ is of the sort, $\exists p\exists q\{\exists v[\mathscr{B}(p,q,\emptyset,v)\wedge\mathscr{G}(\vec{x},v)]\wedge\mathscr{C}\wedge\mathscr{B}(p,q,\emptyset,z)\}$. You need to show, $\mathrm{Q_s}\vdash\forall w[\exists p\exists q\{\exists v[\mathscr{B}(p,q,\emptyset,v)\wedge\mathscr{G}(\overline{\mathsf{m}},v)]\wedge\mathscr{C}\wedge\mathscr{B}(p,q,\emptyset,w)\}\rightarrow w=\overline{\mathsf{k}}_0]$. This is straightforward. So $\mathrm{Q_s}\vdash\forall w[\mathscr{F}(\overline{\mathsf{m}},\emptyset,w)\rightarrow w=\overline{\mathsf{a}}]$.

Assp: $\mathrm{Q_s}\vdash\forall w[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}},w)\rightarrow w=\overline{\mathsf{k}}_n]$

Show: $\mathrm{Q_s}\vdash\forall w[\mathscr{F}(\overline{\mathsf{m}},S\overline{\mathsf{n}},w)\rightarrow w=\overline{\mathsf{k}}_{\mathsf{Sn}}]$

From from capture, $\mathrm{Q_s}\vdash_{ND}\forall w[\mathscr{H}(\overline{\mathsf{m}},\overline{\mathsf{n}},\overline{\mathsf{k}}_n,w)\rightarrow w=\overline{\mathsf{k}}_{\mathsf{Sn}}]$. And again we make an appeal to uniqueness:

1.	$\forall w[\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}},w) \to w = \overline{\mathsf{k}}_{\mathsf{n}}]$	by assumption
2.	$\forall w[\mathscr{H}(\overline{\mathsf{m}},\overline{\mathsf{n}},\overline{\mathsf{k}}_{\mathsf{n}},w) \to w = \overline{\mathsf{k}}_{\mathsf{Sn}}]$	by capture
3.	$\forall p\forall q\forall x\forall y[(\mathscr{B}(p,q,S\overline{\mathsf{n}},x) \wedge \mathscr{B}(p,q,S\overline{\mathsf{n}},y)) \to x = y]$	uniqueness
4.	$\mathscr{F}(\overline{\mathsf{m}},S\overline{\mathsf{n}},j)$	A (g, $\to$I)
5.	$\exists p\exists q[\mathscr{A} \wedge \mathscr{C} \wedge \mathscr{B}]$	4 abv
6.	$\exists q[\mathscr{A} \wedge \mathscr{C} \wedge \mathscr{B}]$	A (g, 5$\exists$E)
7.	$\mathscr{A} \wedge \mathscr{C} \wedge \mathscr{B}$	A (g, 6$\exists$E)
8.	$\exists v[\mathscr{B}(p,q,\emptyset,v) \wedge \mathscr{G}(\overline{\mathsf{m}},v)]$	7 $\wedge$E ($\mathscr{A}$)
9.	$(\forall i < S\overline{\mathsf{n}})\exists u\exists v[\mathscr{B}(p,q,i,u) \wedge \mathscr{B}(p,q,Si,v) \wedge \mathscr{H}(\overline{\mathsf{m}},i,u,v)]$	7 $\wedge$E ($\mathscr{C}$)
10.	$\mathscr{B}(p,q,S\overline{\mathsf{n}},j)$	7 $\wedge$E ($\mathscr{B}$)
11.	$\overline{\mathsf{n}} < S\overline{\mathsf{n}}$	T8.14
12.	$\exists u\exists v[\mathscr{B}(p,q,\overline{\mathsf{n}},u) \wedge \mathscr{B}(p,q,S\overline{\mathsf{n}},v) \wedge \mathscr{H}(\overline{\mathsf{m}},\overline{\mathsf{n}},u,v)]$	9,11 ($\forall$E)
13.	$\exists v[\mathscr{B}(p,q,\overline{\mathsf{n}},u) \wedge \mathscr{B}(p,q,S\overline{\mathsf{n}},v) \wedge \mathscr{H}(\overline{\mathsf{m}},\overline{\mathsf{n}},u,v)]$	A (g, 12$\exists$E)
14.	$\mathscr{B}(p,q,\overline{\mathsf{n}},u) \wedge \mathscr{B}(p,q,S\overline{\mathsf{n}},v) \wedge \mathscr{H}(\overline{\mathsf{m}},\overline{\mathsf{n}},u,v)$	A (g, 13$\exists$E)
15.	$\mathscr{B}(p,q,\overline{\mathsf{n}},u)$	14 $\wedge$E
16.	$(\forall i < \overline{\mathsf{n}})\exists u\exists v[\mathscr{B}(p,q,i,u) \wedge \mathscr{B}(p,q,Si,v) \wedge \mathscr{H}(\overline{\mathsf{m}},i,u,v)]$	9 with T8.21
17.	$\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}},u)$	8,16,15 with $\exists$I
18.	$u = \overline{\mathsf{k}}_{\mathsf{n}}$	1,17 with $\forall$E
19.	$\mathscr{H}(\overline{\mathsf{m}},\overline{\mathsf{n}},u,v)$	14 $\wedge$E
20.	$\mathscr{H}(\overline{\mathsf{m}},\overline{\mathsf{n}},\overline{\mathsf{k}}_{\mathsf{n}},v)$	19,18 =E
21.	$v = \overline{\mathsf{k}}_{\mathsf{Sn}}$	2,20 with $\forall$E
22.	$\mathscr{B}(p,q,S\overline{\mathsf{n}},v)$	14 $\wedge$E
23.	$\mathscr{B}(p,q,S\overline{\mathsf{n}},\overline{\mathsf{k}}_{\mathsf{Sn}})$	22,21 =E
24.	$j = \overline{\mathsf{k}}_{\mathsf{Sn}}$	3,10,23 with $\forall$E
25.	$j = \overline{\mathsf{k}}_{\mathsf{Sn}}$	13,14-24 $\exists$E
26.	$j = \overline{\mathsf{k}}_{\mathsf{Sn}}$	12,13-25 $\exists$E
27.	$j = \overline{\mathsf{k}}_{\mathsf{Sn}}$	6,7-26 $\exists$E
28.	$j = \overline{\mathsf{k}}_{\mathsf{Sn}}$	5,6-27 $\exists$E
29.	$\mathscr{F}(\overline{\mathsf{m}},S\overline{\mathsf{n}},j) \to j = \overline{\mathsf{k}}_{\mathsf{Sn}}$	4-28 $\to$I
30.	$\forall w[\mathscr{F}(\overline{\mathsf{m}},S\overline{\mathsf{n}},w) \to w = \overline{\mathsf{k}}_{\mathsf{Sn}}]$	29 $\forall$I

Lines 8 - 10 of show the content of the assumptions on 4 - 7 which are too long to display in expanded form. Once we are able to show $\mathscr{F}(\overline{\mathsf{m}},\overline{\mathsf{n}},u)$ at (17), the inductive assumption lets us "pin" u onto $\overline{\mathsf{k}}_{\mathsf{n}}$. Then uniqueness conditions for $\mathscr{H}$ and $\mathscr{B}$ allow us to move to unique outputs for $\mathscr{H}$ and $\mathscr{B}$ and so for $\mathscr{F}$. Line 16 perhaps obviously follows from (9), but its derivation may be obscure: by T8.14, Q $\vdash \overline{\mathsf{0}} < S\overline{\mathsf{n}}$ and ... and Q $\vdash \overline{\mathsf{n-1}} < S\overline{\mathsf{n}}$; so where $\mathscr{P}$ is the quantified formula on (9) by ($\forall$E), Q $\vdash \mathscr{P}(\overline{\mathsf{0}})$ and ... and Q $\vdash \mathscr{P}(\overline{\mathsf{n-1}})$;

then by with T8.21 it follows that Q $\vdash (\forall i < \overline{\mathsf{n}})\mathscr{P}(i)$. And we have the theorem by induction.

Indct: For any n, $\mathrm{Q_s} \vdash_{ND} \forall w[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \to w = \overline{\mathsf{k}}_{\mathsf{n}}]$.

Observe that in both the basis and show clauses we require the generalized uniqueness for $\mathscr{B}$: this is because it is being applied inside assumptions for $\exists$E, where p and q are arbitrary variables, not numerals $\overline{\mathsf{p}}$ and $\overline{\mathsf{q}}$, to which the ordinary notion of capture for $\mathscr{B}$ would apply. So $\forall w[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}}, w) \to w = \overline{\mathsf{a}}]$. So we satisfy the recursive clause for T12.11. So the theorem is proved. And we have shown that $\mathrm{Q_s}$ has the resources to capture any recursive function.

This theorem has a number of attractive features: We show that recursive functions are captured directly by the original formulas by which they are expressed. A byproduct is that recursive functions are captured by Σ_1 formulas. The argument is a straightforward induction on the sequence of recursive functions, of a type we have seen before. But we do not show that recursive functions are captured in Q. It is that to which we turn.

*E12.14. Complete the demonstration of T12.11 by completing the remaining cases, including the basis and part (ii) of the case for composition.

*E12.15. Produce a derivation to show the basis of T12.12.

E12.16. Continuing along the lines from E12.7, observe that T12.4 assumes that functions are recursive and so total. In the context of partial functions, CPf would have to be augmented with the condition that if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_\mathsf{n}\rangle, \mathsf{a}\rangle \notin \mathsf{f}$ then $T \vdash \sim\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_\mathsf{n}, \overline{\mathsf{a}})$. Extend the argument for T12.11 to show that on the standard interpretation N for $\mathscr{L}_{\mathrm{NT}}$, any μ-recursive function is captured, on the extended account, in $\mathrm{Q_s}$ by the original formula by which it is expressed.

E12.17. Return to the simple functions from from E12.10. Show that on the standard interpretation N of $\mathscr{L}_{\mathrm{NT}}$ each simple function $\mathsf{f}(\vec{\mathsf{x}})$ is captured in $\mathrm{Q_s}$ by the formula used to express it. Restrict appeal to external theorems just to your result from E12.10 and T8.14 as appropriate.

12.3.3 The result strengthened

T12.11 shows that the recursive functions are captured in $\mathrm{Q_s}$ by their Σ_1 original expressers. As we have suggested, this argument is easily strengthened to show that the recursive functions are captured in Q. To do so, we give up the capture by original expressers, though we retain the result that the recursive functions are captured by Σ_1 formulas.

In the previous section, we appealed to uniqueness of remainder for the β-function. In $\mathrm{Q_s}$, the original formula $\mathscr{B}$ captures the β-function, and gives a strengthened uniqueness result important for T12.12. But we can simulate this effect by some easy theorems. Recall that the β-function is originally expressed by a Δ_0 formula $\mathscr{B}$.

T12.13. If a function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by a Δ_0 formula $\mathscr{F}(\vec{x}, v)$, then $\mathscr{F}'(\vec{x}, v) =_{\text{def}} \mathscr{F}(\vec{x}, v) \wedge (\forall z \leq v)[\mathscr{F}(\vec{x}, z) \rightarrow z = v]$ is Δ_0 and captures f in Q.

Suppose $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by a Δ_0 formula $\mathscr{F}(\vec{x}, v)$ and $\vec{\mathsf{x}}$ reduces to a single variable. Suppose $\langle \mathsf{m}, \mathsf{a}\rangle \in \mathsf{f}$. (a) Then, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$; and since $\mathscr{F}$ is Δ_0, by T12.8, $\mathsf{Q} \vdash_{ND} \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$. (b) Suppose $\mathsf{n} \neq \mathsf{a}$; then $\langle \mathsf{m}, \mathsf{n}\rangle \notin \mathsf{f}$; so with T12.2, $\mathsf{N}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})] = \mathsf{T}$ and $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})] \neq \mathsf{T}$; so by T12.8, $\mathsf{Q} \vdash_{ND} \sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$.

(i) From (a), $\mathsf{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})$. And $\vdash \overline{\mathsf{a}} = \overline{\mathsf{a}}$, so $\vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \rightarrow \overline{\mathsf{a}} = \overline{\mathsf{a}}$; and from (b), for $\mathsf{q} < \mathsf{a}$, $\mathsf{Q} \vdash \sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{q}})$; so trivially, $\mathsf{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{q}}) \rightarrow \overline{\mathsf{q}} = \overline{\mathsf{a}}$; so for any $\mathsf{p} \leq \mathsf{a}$, $\mathsf{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{p}}) \rightarrow \overline{\mathsf{p}} = \overline{\mathsf{a}}$; so by T8.21, $\mathsf{Q} \vdash (\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{a}})$. So with $\wedge$I, $\mathsf{Q} \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \wedge (\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{a}})$; which is to say, $\mathsf{Q} \vdash \mathscr{F}'(\overline{\mathsf{m}}, \overline{\mathsf{a}})$.

(ii) Hint: You need to show $\mathsf{Q} \vdash \forall w([\mathscr{F}(\overline{\mathsf{m}}, w) \wedge (\forall z \leq w)(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow z = w)] \rightarrow w = \overline{\mathsf{a}})$. Take as premises $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}}) \wedge (\forall z \leq \overline{\mathsf{a}})(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{a}})$ from (i), along with $j \leq \overline{\mathsf{a}} \vee \overline{\mathsf{a}} \leq j$ from T8.19.

This result effectively tells us that if conditions (a) and (b) are met, then there is an $\mathscr{F}'$ that captures f. This $\mathscr{F}'$ is not the same as the original $\mathscr{F}$ that expresses the function. Still, if the Δ_0 $\mathscr{B}$ expresses the β-function, we have $\mathscr{B}'$ that captures it in Q. Intuitively, the second conjunct of $\mathscr{F}'$ tells us that any $z < v$ cannot satisfy $\mathscr{F}$.

Further, formulas of the sort $\mathscr{F}'$ yield a modified uniqueness result.

T12.14. For $\mathscr{F}'(\vec{x}, v) =_{\text{def}} \mathscr{F}(\vec{x}, v) \wedge (\forall z \leq v)[\mathscr{F}(\vec{x}, z) \rightarrow z = v]$ as above, for any n, $\mathsf{Q} \vdash \forall \vec{x} \forall y[(\mathscr{F}'(\vec{x}, \overline{\mathsf{n}}) \wedge \mathscr{F}'(\vec{x}, y)) \rightarrow y = \overline{\mathsf{n}}]$. Suppose $\vec{x}$ reduces to a single variable and reason as follows,

1.	$\forall x(x \leq \overline{\mathsf{n}} \vee \overline{\mathsf{n}} \leq x)$	T8.19
2.	$\mathscr{F}'(j, \overline{\mathsf{n}}) \wedge \mathscr{F}'(j, k)$	A (g $\rightarrow$I)
3.	$\mathscr{F}(j, \overline{\mathsf{n}}) \wedge (\forall z \leq \overline{\mathsf{n}})(\mathscr{F}(j, z) \rightarrow z = \overline{\mathsf{n}})$	2 $\wedge$E (unabv)
4.	$\mathscr{F}(j, k) \wedge (\forall z \leq k)(\mathscr{F}(j, z) \rightarrow z = k)$	2 $\wedge$E (unabv)
5.	$k \leq \overline{\mathsf{n}} \vee \overline{\mathsf{n}} \leq k$	1 $\forall$E
6.	$k \leq \overline{\mathsf{n}}$	A (g 5$\vee$E)
7.	$(\forall z \leq \overline{\mathsf{n}})(\mathscr{F}(j, z) \rightarrow z = \overline{\mathsf{n}})$	3 $\wedge$E
8.	$\mathscr{F}(j, k) \rightarrow k = \overline{\mathsf{n}}$	7,6 ($\forall$)E
9.	$\mathscr{F}(j, k)$	4 $\wedge$E
10.	$k = \overline{\mathsf{n}}$	8,9 $\rightarrow$E
11.	$\overline{\mathsf{n}} \leq k$	A (g 5$\vee$E)
	$\vdots$	
12.	$k = \overline{\mathsf{n}}$	
13.	$k = \overline{\mathsf{n}}$	5,6-10,11-12 $\vee$E
14.	$(\mathscr{F}'(j, \overline{\mathsf{n}}) \wedge \mathscr{F}'(j, k)) \rightarrow k = \overline{\mathsf{n}}$	2-13 $\rightarrow$I
15.	$\forall y[(\mathscr{F}'(j, \overline{\mathsf{n}}) \wedge \mathscr{F}'(j, y)) \rightarrow y = \overline{\mathsf{n}}]$	14 $\forall$I
16.	$\forall x \forall y[(\mathscr{F}'(x, \overline{\mathsf{n}}) \wedge \mathscr{F}'(x, y)) \rightarrow y = \overline{\mathsf{n}}]$	15 $\forall$I

Reasoning for the second subderivation is similar to the first.

So where p, q and v are universally quantified we shall have, $\mathsf{Q} \vdash \forall p \forall q \forall v[(\mathscr{B}'(p, q, \overline{\mathsf{m}}, \overline{\mathsf{n}}) \wedge \mathscr{B}'(p, q, \overline{\mathsf{m}}, v)) \rightarrow v = \overline{\mathsf{n}}]$. Because $\overline{\mathsf{n}}$ is a numeral, this is not quite what we had from $\mathrm{Q_s}$, but it it will be sufficient for what we want.

Observe also that insofar as $\mathscr{F}'(\vec{x}, v)$ is built on an $\mathscr{F}(\vec{x}, v)$ that expresses a function, $\mathscr{F}'(\vec{x}, v)$ continues to expresses $\mathsf{f}(\vec{\mathsf{x}})$. Perhaps this is obvious given what $\mathscr{F}'$ says. However, we can argue for the result directly.

T12.15. If $\mathscr{F}(\vec{x}, v)$ expresses $\mathsf{f}(\vec{\mathsf{x}})$, then $\mathscr{F}'(\vec{x}, v) = \mathscr{F}(\vec{x}, v) \wedge (\forall z \leq v)[\mathscr{F}(\vec{x}, z) \rightarrow z = v]$ expresses $\mathsf{f}(\vec{\mathsf{x}})$.

Suppose $\vec{\mathsf{x}}$ reduces to a single variable and $\mathsf{f}(\mathsf{x})$ is expressed by $\mathscr{F}(x, v)$. Suppose $\langle \mathsf{m}, \mathsf{a} \rangle \in \mathsf{f}$. (a) By expression, $\mathsf{N}[\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{a}})] = \mathsf{T}$. (b) Suppose $\mathsf{n} \neq \mathsf{a}$; then $\langle \mathsf{m}, \mathsf{n} \rangle \notin \mathsf{f}$; so with T12.2, $\mathsf{N}[\sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})] = \mathsf{T}$.

(i) Suppose $\mathsf{N}[\mathscr{F}'(\overline{\mathsf{m}}, \overline{\mathsf{a}})] \neq \mathsf{T}$. This is impossible. You will need applications of T12.6 and T10.2; observe that for $\mathsf{n} \leq \mathsf{a}$ either $\mathsf{n} = \mathsf{a}$ or $\mathsf{n} < \mathsf{a}$ (so that $\mathsf{n} \neq \mathsf{a}$).

(ii) Suppose $\mathsf{N}[\forall w([\mathscr{F}(\overline{\mathsf{m}}, w) \wedge (\forall z \leq w)(\mathscr{F}(\overline{\mathsf{m}}, z) \rightarrow z = w)] \rightarrow w = \overline{\mathsf{a}})] \neq \mathsf{T}$. This is impossible. This time, you will be able to reason that for any n either $\mathsf{n} = \mathsf{a}$ or $\mathsf{n} \neq \mathsf{a}$.

And now we are in a position to recover the main result, except that the recursive functions are captured in Q rather than Q_s.

T12.16. Any recursive function is captured by a Σ_1 formula in Q

The β-function is expressed by a Δ_0 formula $\mathscr{B}(p, q, i, v)$; so by T12.15 and T12.13 there is a Δ_0 formula $\mathscr{B}'(p, q, i, v)$ that expresses and captures it in Q. For any $\mathsf{f}(\vec{\mathsf{x}})$ originally expressed by $\mathscr{F}(\vec{x}, v)$, let $\mathscr{F}^{\dagger}$ be like $\mathscr{F}$ except that instances of $\mathscr{B}$ are replaced by $\mathscr{B}'$. Since $\mathscr{B}'$ is Δ_0, $\mathscr{F}^{\dagger}$ remains Σ_1.

The argument is now a matter of showing that demonstrations of T12.3, T12.11 and T12.12 go through with application to these formulas and in Q. For the first two, the argument is nearly trivial: everything is the same as before with formulas of the sort $\mathscr{F}^{\dagger}$ replacing $\mathscr{F}$. For the last, it will be important that derivations which rely on uniqueness for the β-function go through with the result from T12.14, that for any m and n, $\mathrm{Q} \vdash \forall p \forall q \forall v[(\mathscr{B}'(p, q, \overline{\mathsf{m}}, \overline{\mathsf{n}}) \wedge \mathscr{B}'(p, q, \overline{\mathsf{m}}, v)) \rightarrow v = \overline{\mathsf{n}}]$.

Be clear that expressions of the sort $\mathscr{F}^{\dagger}$ might appear all along in the show part of T12.3, T12.11 and T12.12. Expressions from the basis do not involve $\mathscr{B}$. It is included by recursion; after that, composition and regular minimization might be applied to expressions of any sort, and so to ones which involve $\mathscr{B}$ as well.

As in for the case of expression, formulas other than $\mathscr{F}^{\dagger}(\vec{x}, v)$ might capture the recursive functions — for example, if $\mathscr{F}^{\dagger}(\vec{x}, v)$ captures $\mathsf{f}(\vec{\mathsf{x}})$, then so does $\mathscr{F}^{\dagger}(\vec{x}, v) \wedge \mathscr{A}$ for any theorem $\mathscr{A}$. Let us say that $\mathscr{F}^{\dagger}(\vec{x}, v)$ is the *canonical* formula that captures $\mathsf{f}(\vec{\mathsf{x}})$ in Q. Of course, the canonical formula which captures $\mathsf{f}(\vec{\mathsf{x}})$ need not be the same as the corresponding original formula — for the β-function is not captured by its original formula (and so any formula which includes a β-function fails to be original). Because the β-function is captured by a Δ_0 formula we do, however, retain the result that every recursive function is captured in Q by some Σ_1 formula.

For the following, unless otherwise noted, when on the basis of our theorems, we assert the existence of a formula to express or some capture recursive function, we shall have in mind the *canonical* formula. Thus a function is expressed and captured by the same formula.

E12.18. Provide an argument to demonstrate (ii) of T12.13.

E12.19. Finish the derivation for T12.14 by completing the second subderivation.

E12.20. Complete the demonstration of T12.15.

*E12.21. Work carefully through the demonstration of T12.16 by setting up revised arguments T12.3†, T12.11† and T12.12†. As feasible, you may simply explain how parts differ from the originals. For the last, be sure that derivations work with revised uniqueness conditions.

12.4 More Recursive Functions

Now that we have seen what the recursive functions are, and the powers of our logical systems to express and capture recursive functions, we turn to extending their range. In fact, in this section, we shall generate a series of functions that are *primitive* recursive. In addition to the initial functions, so far, we have seen that plus, times, fact and power are primitive recursive. As we increase the range of (primitive) recursive functions, it immediately follows that our logical systems have the power to express and capture all the same functions.

12.4.1 Preliminary Functions

We begin with some simple primitive recursive functions that will serve as a foundation for things to come.

Predecessor with cutoff. Set the predecessor of zero to zero itself, and for any other value to the one before. Since $\mathsf{pred}(\mathsf{y})$ is a one-place function, gpred is a constant, in this case, $\mathsf{gpred} = 0$. And $\mathsf{hpred} = \mathsf{idnt}^2_1(\mathsf{y}, \mathsf{u})$. So, as we expect for $\mathsf{pred}(\mathsf{y})$,

$$\begin{aligned}&\mathsf{pred}(0) = 0\\&\mathsf{pred}(\mathsf{suc}(\mathsf{y})) = \mathsf{y}\end{aligned}$$

So predecessor is a primitive recursive function.

Subtraction with cutoff. When $\mathsf{y} \geq \mathsf{x}$, $\mathsf{subc}(\mathsf{x}, \mathsf{y}) = 0$. Otherwise $\mathsf{subc}(\mathsf{x}, \mathsf{y}) = \mathsf{x} - \mathsf{y}$. For $\mathsf{subc}(\mathsf{x}, \mathsf{y})$, set $\mathsf{gsubc}(\mathsf{x}) = \mathsf{idnt}^1_1(\mathsf{x})$. And $\mathsf{hsubc}(\mathsf{x}, \mathsf{y}, \mathsf{u}) = \mathsf{pred}(\mathsf{idnt}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{u}))$. So,

$$\begin{aligned}&\mathsf{subc}(\mathsf{x}, 0) = \mathsf{x}\\&\mathsf{subc}(\mathsf{x}, \mathsf{suc}(\mathsf{y})) = \mathsf{pred}(\mathsf{subc}(\mathsf{x}, \mathsf{y}))\end{aligned}$$

So as y increases by one, the difference decreases by one. Informally, indicate $\mathsf{subc}(\mathsf{x}, \mathsf{y}) = (\mathsf{x} \mathbin{\dot{-}} \mathsf{y})$.

Absolute value. $\mathsf{absval}(\mathsf{x} - \mathsf{y}) = (\mathsf{x} \dot{-} \mathsf{y}) + (\mathsf{y} \dot{-} \mathsf{x})$. So we find the absolute value of the difference between x and y by doing the subtraction with cutoff both ways. One direction yields zero. The other yields the value we want. So the sum comes out to the absolute value. This is a function with two arguments (only separated by '-' rather than comma to remind us of the nature of the function). This function results entirely by composition, without a recursion clause. Informally, we indicate absolute value in the usual way, $\mathsf{absval}(\mathsf{x} - \mathsf{y}) = |\mathsf{x} - \mathsf{y}|$.

Sign. The function $\mathsf{sg}(\mathsf{y})$ is zero when y is zero and otherwise one. For $\mathsf{sg}(\mathsf{y})$, set $\mathsf{gsg} = 0$. And $\mathsf{hsg}(\mathsf{y}, \mathsf{u}) = \mathsf{suc}(\mathsf{zero}(\mathsf{idnt}^2_1(\mathsf{y}, \mathsf{u})))$. So,

$$\begin{aligned}&\mathsf{sg}(0) = 0\\&\mathsf{sg}(\mathsf{suc}(\mathsf{y})) = \mathsf{suc}(\mathsf{zero}(\mathsf{y}))\end{aligned}$$

So the sign of any successor is just the successor of zero, which is one.

Converse sign. The function $\mathsf{csg}(\mathsf{y})$ is one when y is zero and otherwise zero. So it inverts sg. For $\mathsf{csg}(\mathsf{y})$, set $\mathsf{gcsg} = \mathsf{suc}(0)$. And $\mathsf{hcsg}(\mathsf{y}, \mathsf{u}) = \mathsf{zero}(\mathsf{idnt}^2_1(\mathsf{y}, \mathsf{u}))$. So,

$$\begin{aligned}&\mathsf{csg}(0) = \mathsf{suc}(0)\\&\mathsf{csg}(\mathsf{suc}(\mathsf{y})) = \mathsf{zero}(\mathsf{y})\end{aligned}$$

So the converse sign of any successor is just zero. Informally, we indicate the converse sign with a bar, $\overline{\mathsf{sg}}(\mathsf{y})$.

E12.22. Consider again your file `recursive1.rb` from E12.3. Extend your sequence of functions to include `pred(x)`, `subc(x,y)`, `absval(x - y)`, `sg(x)`, and `csg(x)`. Calculate some values of these functions and print the results, along with your program. Again, there should be no appeal to functions except from earlier in the chain.

12.4.2 Characteristic Functions

(CF) For any function $\mathsf{p}(\vec{\mathsf{x}})$, $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}}))$ is the *characteristic* function of the relation R such that $\vec{\mathsf{x}} \in \mathsf{R}$ iff $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}})) = 0$. So a characteristic function for relation R takes just the values 0 and 1 and if $\mathsf{R}(\vec{\mathsf{x}})$ is true, then $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}}) = 0$ and if $\mathsf{R}(\vec{\mathsf{x}})$ is not true, then $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}}) = 1$.[6] A (*primitive*) *recursive* property or relation is one that has a (primitive) recursive characteristic function — though when p already takes just the values 0

[6] It is perhaps more common to reverse the values of zero and one for the characteristic function. However, the choice is arbitrary, and this choice is technically convenient.

and 1 so that $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}})) = \mathsf{p}(\vec{\mathsf{x}})$, we generally omit sg from our specifications. These definitions immediately result in corollaries to T12.3 and T12.16.

T12.3 (corollary). On the standard interpretation N of $\mathscr{L}_{NT}$, each recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ is expressed by some formula $\mathcal{R}(\vec{x})$.

> Suppose $\mathsf{R}(\vec{\mathsf{x}})$ is a recursive relation; then it has a recursive characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$; so by T12.3 there is some formula $\mathcal{R}(\vec{x}, y)$ that expresses $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$. So in the case where $\vec{x}$ reduces to a single variable, if $\mathsf{m} \in \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \in \mathsf{ch}_{\mathsf{R}}$; and by expression, $\mathsf{I}[\mathcal{R}(\overline{\mathsf{m}}, \emptyset)] = \mathsf{T}$; and if $\mathsf{m} \notin \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \notin \mathsf{ch}_{\mathsf{R}}$, so that with T12.2, $\mathsf{I}[\sim\mathcal{R}(\overline{\mathsf{m}}, \emptyset)] = \mathsf{T}$. So, generally, $\mathcal{R}(\vec{x}, \emptyset)$ expresses $\mathsf{R}(\vec{\mathsf{x}})$.

T12.16 (corollary). Any recursive relation is captured by a Σ_1 formula in Q.

> Suppose $\mathsf{R}(\vec{\mathsf{x}})$ is a recursive relation; then it has a recursive characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$; so by T12.16 there is some Σ_1 formula $\mathcal{R}(\vec{x}, y)$ that captures $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$. So in the case where $\vec{x}$ reduces to a single variable, if $\mathsf{m} \in \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \in \mathsf{ch}_{\mathsf{R}}$; and by capture $T \vdash \mathcal{R}(\overline{\mathsf{m}}, \emptyset)$; and if $\mathsf{m} \notin \mathsf{R}$, then $\langle \mathsf{m}, 0\rangle \notin \mathsf{ch}_{\mathsf{R}}$; so by capture with T12.4, $T \vdash \sim\mathcal{R}(\overline{\mathsf{m}}, \emptyset)$. So, generally $\mathcal{R}(\vec{x}, \emptyset)$ captures $\mathsf{R}(\vec{\mathsf{x}})$.

So our results for the expression and capture of recursive functions extend directly to the expression and capture of recursive relations: a recursive relation has a recursive characteristic function; as such, the function is expressed and captured; so, as we have just seen, the corresponding relation is expressed and captured.

Equality. Say $\mathsf{t}(\vec{\mathsf{x}})$ is a *recursive term* just in case it is a variable, constant, or a recursive function. Then for any recursive terms $\mathsf{s}(\vec{\mathsf{x}})$ and $\mathsf{t}(\vec{\mathsf{y}})$, $\mathsf{EQ}(\mathsf{s}(\vec{\mathsf{x}}), \mathsf{t}(\vec{\mathsf{y}}))$ — typically rendered $\mathsf{s}(\vec{\mathsf{x}}) = \mathsf{t}(\vec{\mathsf{y}})$, is a recursive relation with characteristic function $\mathsf{ch}_{\mathsf{EQ}}(\vec{\mathsf{x}}, \vec{\mathsf{y}}) = \mathsf{sg}|\mathsf{s}(\vec{\mathsf{x}}) - \mathsf{t}(\vec{\mathsf{y}})|$. When $\mathsf{s}(\vec{\mathsf{x}})$ is equal to $\mathsf{t}(\vec{\mathsf{y}})$, the absolute value of the difference is zero so the value of sg is zero. But when $\mathsf{s}(\vec{\mathsf{x}})$ is other than $\mathsf{t}(\vec{\mathsf{y}})$, the absolute value of the difference is other than zero, so value of sg is one. And, supposing that $\mathsf{s}(\vec{\mathsf{x}})$ and $\mathsf{t}(\vec{\mathsf{x}})$ are recursive, this characteristic function is a composition of recursive functions. So the result is recursive. So $\mathsf{s}(\vec{\mathsf{x}}) = \mathsf{t}(\vec{\mathsf{y}})$ is a recursive relation.

A couple of observations: First, be clear that EQ is the standard relation we all know and love. The trick is to show that it is recursive. We are not *given* that EQ is a recursive relation — so we demonstrate that it is, by showing that it has a recursive characteristic function. Second, one might think that we could express $\mathsf{f}(\vec{\mathsf{x}}) = \mathsf{g}(\vec{\mathsf{y}})$ by some relatively simple expression that would compose expressions for the functions

with equality as, $\exists u \exists v[\mathscr{F}(\vec{x}, u) \wedge \mathscr{G}(\vec{y}, v) \wedge u = v)$. This would be fine. However we have offered a general account which, as is often the case for these things, need not be the most efficient. Where $\mathsf{sg}|\mathsf{f}(\vec{\mathsf{x}}) - \mathsf{g}(\vec{\mathsf{y}})|$ is expressed and captured by some $\mathscr{S}(\vec{x}, \vec{y}, v)$ our approach, which works by modification of the characteristic function, generates the relatively complex, $\mathscr{E}(\vec{x}, \vec{y}) =_{\text{def}} \mathscr{S}(\vec{x}, \vec{y}, \emptyset)$.

Inequality. The relation $\textsc{leq}(\mathsf{s}(\vec{\mathsf{x}}), \mathsf{t}(\vec{\mathsf{y}}))$ has characteristic function $\mathsf{sg}(\mathsf{s}(\vec{\mathsf{x}}) \dot{-} \mathsf{t}(\vec{\mathsf{y}}))$. When $\mathsf{s}(\vec{\mathsf{x}}) \leq \mathsf{t}(\vec{\mathsf{y}})$, $\mathsf{s}(\vec{\mathsf{x}}) \dot{-} \mathsf{t}(\vec{\mathsf{y}}) = 0$; so $\mathsf{sg} = 0$; Otherwise the value is 1. The relation $\textsc{less}(\mathsf{s}(\vec{\mathsf{x}}), \mathsf{t}(\vec{\mathsf{y}}))$ has characteristic function $\mathsf{sg}(\mathsf{suc}(\mathsf{s}(\vec{\mathsf{x}})) \dot{-} \mathsf{t}(\vec{\mathsf{y}}))$. When $\mathsf{s}(\vec{\mathsf{x}}) < \mathsf{t}(\vec{\mathsf{y}})$, $\mathsf{suc}(\mathsf{s}(\vec{\mathsf{x}})) \dot{-} \mathsf{t}(\vec{\mathsf{y}}) = 0$; so $\mathsf{sg} = 0$. Otherwise the value is 1. These are typically represented $\mathsf{s}(\vec{\mathsf{x}}) \leq \mathsf{t}(\vec{\mathsf{y}})$ and $\mathsf{s}(\vec{\mathsf{x}}) < \mathsf{t}(\vec{\mathsf{y}})$.

With equality and inequality, we have atomic recursive relations. And we set out to exhibit ones that are more complex in the usual way.

Truth functions. Suppose $\mathsf{P}(\vec{\mathsf{x}})$ and $\mathsf{Q}(\vec{\mathsf{x}})$ are recursive relations. Then $\textsc{neg}(\mathsf{P}(\vec{\mathsf{x}}))$ and $\textsc{dsj}(\mathsf{P}(\vec{\mathsf{x}}), \mathsf{Q}(\vec{\mathsf{x}}))$ are recursive relations. Suppose $\mathsf{ch}_{\mathsf{P}}(\vec{\mathsf{x}})$ and $\mathsf{ch}_{\mathsf{Q}}(\vec{\mathsf{x}})$ are the characteristic functions of $\mathsf{P}(\vec{\mathsf{x}})$ and $\mathsf{Q}(\vec{\mathsf{x}})$.

$\textsc{neg}(\mathsf{P}(\vec{\mathsf{x}}))$ (typically $\sim\mathsf{P}(\vec{\mathsf{x}})$) has characteristic function $\overline{\mathsf{sg}}(\mathsf{ch}_{\mathsf{P}}(\vec{\mathsf{x}}))$. When $\mathsf{P}(\vec{\mathsf{x}})$ does not obtain, the characteristic function of $\mathsf{P}(\vec{\mathsf{x}})$ takes value one, so the converse sign goes to zero. And when when $\mathsf{P}(\vec{\mathsf{x}})$ does obtain, its characteristic function is zero, so the converse sign is one — which is as it should be.

$\textsc{dsj}(\mathsf{P}(\vec{\mathsf{x}}), \mathsf{Q}(\vec{\mathsf{y}}))$ (typically $\mathsf{P}(\vec{\mathsf{x}}) \vee \mathsf{Q}(\vec{\mathsf{y}})$) has characteristic function $\mathsf{ch}_{\mathsf{P}}(\vec{\mathsf{x}}) \times \mathsf{ch}_{\mathsf{Q}}(\vec{\mathsf{y}})$. When one of $\mathsf{P}(\vec{\mathsf{x}})$ or $\mathsf{Q}(\vec{\mathsf{y}})$ is true, the disjunction is true; but in this case, at least one characteristic function, and so the product of functions goes to zero. If neither $\mathsf{P}(\vec{\mathsf{x}})$ nor $\mathsf{Q}(\vec{\mathsf{y}})$ is true, the disjunction is not true; in this case, both characteristic functions, and so the product of functions take the value one.

Other truth functions are definable in the same terms as for negation and disjunction. So, for example, $\textsc{imp}(\mathsf{P}(\vec{\mathsf{x}}), \mathsf{Q}(\vec{\mathsf{y}}))$ that is, $\mathsf{P}(\vec{\mathsf{x}}) \rightarrow \mathsf{Q}(\vec{\mathsf{y}})$ is just $\sim\mathsf{P}(\vec{\mathsf{x}}) \vee \mathsf{Q}(\vec{\mathsf{y}})$.

Bounded quantifiers: Consider a relation $\mathsf{S}(\vec{\mathsf{x}}, \mathsf{z}) = (\exists \mathsf{y} \leq \mathsf{z})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$ which obtains when there is a y less than or equal to z such that $\mathsf{P}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$. The variable z for the bound may or may not have a natural place in P, though we treat it as at least a placeholder insofar as it has a definite place in $\mathsf{s}(\vec{\mathsf{x}}, \mathsf{z})$. Given $\mathsf{ch}_{\mathsf{P}}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$, consider a further relation $\mathsf{R}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{v})$ corresponding to $(\exists \mathsf{y} \leq \mathsf{v})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$. So R treats the bound as a separate variable, and will let us reason by induction as the bound ranges from 0 to z. If we can find $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{v})$ then $\mathsf{ch}_{\mathsf{S}}(\vec{\mathsf{x}}, \mathsf{z})$ is automatic as $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{z})$. For this $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{v})$ set,

$gch_R(\vec{x}, z) = ch_P(\vec{x}, z, 0)$
$hch_R(\vec{x}, z, y, u) = u \times ch_P(\vec{x}, z, Sy)$

In the simple case where $\vec{x}$ drops out, $ch_R(z, 0) = ch_P(z, 0)$. And $ch_R(z, Sy) = ch_R(z, y) \times ch_P(z, Sy)$. The result is,

$ch_R(z, v) = ch_P(z, 0) \times ch_P(z, 1) \times \ldots \times ch_P(z, v)$

Think of these as grouped to the left. So the result has $ch_R(z, n) = 1$ unless and until one of the members is zero, and then stays zero. So the function for $R(y, v)$ goes to zero just in case $P(z, y)$ is true for some value between 0 and v. So set $ch_S(\vec{x}, z) = ch_R(\vec{x}, z, z)$ — so the characteristic function for the bounded quantifier runs the R function up to the bound z.

For $\overset{<}{S}(\vec{x}, z) = (\exists y < z)P(\vec{x}, z, y)$, adopt $\overset{<}{R}(\vec{x}, z, v)$ for $(\exists y < v)P(\vec{x}, z, y)$ with $ch_{\overset{<}{R}}(\vec{x}, z, v)$ such that $gch_{\overset{<}{R}}(\vec{x}, z) = suc(zero(ch_P(\vec{x}, z, 0)))$; so that $ch_{\overset{<}{R}}(\vec{x}, z, 0) = 1$; since there is no y less than zero such that $P(\vec{x}, z, y)$, $ch_{\overset{<}{R}}$ goes automatically to one. And set $hch_{\overset{<}{R}}(\vec{x}, z, y, u) = u \times ch_P(\vec{x}, z, y)$; so in the simple case, $ch_{\overset{<}{R}}(z, Sy) = ch_{\overset{<}{R}}(z, y) \times ch_P(z, y)$, and we check only values prior to Sy. Then as before, $ch_{\overset{<}{S}}(\vec{x}, z) = ch_{\overset{<}{R}}(\vec{x}, z, z)$.

For $(\forall z \leq y)P(\vec{x}, z)$ and $(\forall z < y)P(\vec{x}, z)$, it is simplest just to consider $\sim(\exists z \leq y)\sim P(\vec{x}, z)$; and similarly in the other case. And we are done by previous results.

Least element: Let $m(\vec{x}, z) = (\mu y \leq z)P(\vec{x}, z, y)$ be the least $y \leq z$ such that $P(\vec{x}, z, y)$ if one exists, and otherwise z. Then if $P(\vec{x}, z, y)$ is a recursive relation, $(\mu y \leq z)P(\vec{x}, z, y)$ is a recursive function. First take $R(\vec{x}, z, v)$ for $(\exists y \leq v)P(\vec{x}, z, y)$ and $ch_R(\vec{x}, z, v)$ as described above. So $ch_R(\vec{x}, z, v)$ goes to 0 when P is true for some $j \leq v$. Then, second, adopt a function $q(\vec{x}, z, v)$ corresponding to $(\mu y \leq v)P(\vec{x}, z, y)$. Given this, very much as before, $m(\vec{x}, z)$ is automatic as $q(\vec{x}, z, z)$. For $q(\vec{x}, z, v)$ set,

$gq(\vec{x}, z) = zero(ch_R(\vec{x}, z, 0))$
$hq(\vec{x}, z, y, u) = u + ch_R(\vec{x}, z, y)$

So in the simple case where $\vec{x}$ drops out, $q(z, 0) = 0$; for the least $z \leq 0$ that satisfies any P can only be 0. And then $q(z, Sy) = q(z, y) + ch_R(z, y)$. The result is,

$q(z, Sn) = 0 + ch_R(z, 0) + \ldots + ch_R(z, n)$

where ch_R is 1 until it hits a member that is P and then goes to 0 and stays there. Observe that since this series starts with $y = 0$ and ends with $y = n$ (excluding the first member) it has Sn members; so if all the values are 1 it evaluates to Sn. If there is some a such that $ch_R(z, a)$ is zero, then all the members prior to it are 1 and the sum is a. So set $m(\vec{x}, z) = q(\vec{x}, z, z)$, so that we take the sum up to the limit z. Observe

that $(\mu y \le z)\mathrm{P}(\vec{x}, z, y) = z$ does not require that $\mathrm{P}(\vec{x}, z, z)$ — only that no $a < z$ is such that $\mathrm{P}(\vec{x}, z, a)$.

Selection by cases. Suppose $f_0(\vec{x}) \ldots f_k(\vec{x})$ are recursive functions and $c_0(\vec{x}) \ldots c_k(\vec{x})$ are mutually exclusive recursive relations. Then $f(\vec{x})/c_0 \ldots c_k$ defined as follows is recursive.

$$f(\vec{x}) = \begin{cases} f_0(\vec{x}) \text{ if } c_0(\vec{x}) \\ f_1(\vec{x}) \text{ if } c_1(\vec{x}) \\ \vdots \\ f_k(\vec{x}) \text{ if } c_k(\vec{x}) \\ \text{and otherwise } a \end{cases}$$

Observe that, $f(\vec{x}) =$

$$[\overline{sg}(ch_{c_0}(\vec{x})) \times f_0(\vec{x}) + \overline{sg}(ch_{c_1}(\vec{x})) \times f_1(\vec{x}) + \ldots + \overline{sg}(ch_{c_k}(\vec{x})) \times f_k(\vec{x})] + [ch_{c_0}(\vec{x}) \times ch_{c_1}(\vec{x}) \times \ldots \times ch_{c_k}(\vec{x}) \times a]$$

works as we want. Each of the first terms in this sum is 0 unless the c_i is met in which case $\overline{sg}(ch_{c_i}(\vec{x}))$ is 1 and the term goes to $f_i(\vec{x})$. The final term is 0 unless no condition c_i is met, in which case it is a. So $f(\vec{x})$ is a composition of recursive functions, and itself recursive.

We turn now to some applications that will be particularly useful for things to come. In many ways, the project is like a cool translation exercise — pitched at the level of functions.

Factor. Let $\mathrm{FCTR}(m, n)$ be the relation that obtains between m and n when $m + 1$ evenly divides n (typically, $m|n$). Division is by $m+1$ to avoid worries about division by zero.[7] Then $m|n$ is recursive. This relation is defined as follows.

$$(\exists y \le n)(Sm \times y = n)$$

Observe that this makes (the predecessor of) both 1 and n factors of n, and any number a factor of zero. Since each part is recursive, the whole is recursive. The argument is from the parts to the whole: $Sm \times y = n$ has a recursive characteristic function; so the bounded quantification has a recursive characteristic function; so the factor relation is recursive.

[7] In fact, this is a (minor) complication at this stage, but it will be helpful down the road. See p. 633n10.

Prime number. Say $\text{PRIME}(n)$ is true just when n is a prime number. This property is defined as follows.

$$n > 1 \wedge (\forall j < n)[j|n \rightarrow (Sj = \overline{1} \vee Sj = n)]$$

So n is greater than 1 and the successor of any number that divides it is either $\overline{1}$ or n itself.

Prime sequence. Say the primes are π_0, $\pi_1 \ldots$. Let the value of the function $pi(n)$ (usually $\boldsymbol{\pi}(n)$) be π_n. Then $\boldsymbol{\pi}(n)$ is defined by recursion as follows.

$gpi = suc(suc(0))$
$hpi(y, u) = (\mu y \leq u! + 1)(u < y \wedge \text{PRIME}(y))$

So the first prime, $\boldsymbol{\pi}(0) = 2$. And $\boldsymbol{\pi}(Sn) = (\mu z \leq \boldsymbol{\pi}(n)! + 1)(\boldsymbol{\pi}(n) < y \wedge \text{PRIME}(y))$. So at any stage, the next prime is the least prime which is greater than $\boldsymbol{\pi}(n)$. This depends on the point that all the primes $\leq \pi_n$ are included in the product $\boldsymbol{\pi}(n)!$ Let $p(n) = \pi_0 \times \pi_1 \times \ldots \times \pi_n$. By a standard argument (see G2 in the arithmetic for Gödel numbering reference, p. 477), $p(n) + 1$ is not divisible by any of the primes up to π_n; so either $p(n) + 1$ is itself prime, or there is some prime greater than π_n but less than $p(n) + 1$. But since $\boldsymbol{\pi}(n)!$ is a product including all the primes up to π_n, $p(n) \leq \boldsymbol{\pi}(n)!$; so either $\boldsymbol{\pi}(n)! + 1$ is prime or there is a prime greater than π_n but less than $\boldsymbol{\pi}(n)! + 1$ — and the next prime is sure to appear in the specified range.

Prime exponent. Let $exp(n, i)$ be the (possibly 0) exponent of π_i in the unique prime factorization of n. Then $exp(n, i)$ is recursive. This function may be defined as follows.

$$(\mu x \leq n)[pred(\pi_i^x)|n \wedge \sim pred(\pi_i^{x+1})|n]$$

And, of course, π_i is just $\boldsymbol{\pi}(i)$. Observe that no exponent in the prime factorization of n is greater than n itself — for any $x \geq 2$, $x^n \geq n$ — so the bound is safe. This function returns the first x such that π_i^x divides n but π_i^{x+1} does not.

Prime length. Say a prime π_a is *included* in the factorization of n just in case $a \leq b$ and for some exponent $e_b > 0$, (the predecessor of) $\pi_b^{e_b}$ is a factor of n. So we think of a prime factorization as,

$$\pi_0^{e_0} \times \pi_1^{e_1} \times \ldots \times \pi_b^{e_b}$$

where $e_b > 0$, but exponents for prior members of the series may be zero or not. Then $\mathsf{len}(n)$ is the number of primes included in the prime factorization of n; so $\mathsf{len}(0) = \mathsf{len}(1) = 0$ and otherwise, since the series of primes begins with zero, $\mathsf{len}(n) = b + 1$. For this set,

$$\mathsf{len}(n) =_{\text{def}} (\mu y \leq n)(\forall z : y \leq z \leq n)\mathsf{exp}(n, z) = 0$$

Officially: $(\mu y \leq n)(\forall z \leq n)[z \geq y \rightarrow \mathsf{exp}(n, z) = 0]$. So we find the least y such that none of the primes between π_y and π_n are part of the factorization of n; but then all of the primes prior to it are members of the factorization so that y numbers the length of the factorization. This depends on its being the case that $n < \pi_n$ so that π_n is never included in the factorization of n.

E12.23. Returning to your file `recursive1.rb` from E12.3 and E12.22, extend the sequence of functions to include the characteristic function for $\mathrm{FCTR}(m, n)$. You will need to begin with `cheq(a,b)` for the characteristic function of $a = b$ and then the characteristic function of $Sm \times y = n$. Then you will require a function like $\mathsf{ch}_R(m, n, v)$ corresponding to $(\exists y \leq v)(Sm \times y = n)$. Calculate some values of these functions and print the results, along with your program.

E12.24. Continue in your file `recursive1.rb` to build the characteristic function for $\mathrm{PRIME}(n)$. You will have to build gradually to this result (treating the existential quantifier as primitive so that the universal quantifier appears as $\sim(\exists j < n)\sim P$). You will need `chless(a,b)` and then `chneg(a)`, `chdsj(a,b)`, `chimp(a,b)`, and `chand(a,b)` for the relevant truth functions. With these in hand, you can build a function `chp(n,j)` corresponding to $\sim(j | n \rightarrow (j = 0 \vee j = n))$. And with that, you can obtain a function like $\overset{<}{R}(n, j, v)$ and then the characteristic function of the bounded existential. Then, finally, build `prime(n)`. Calculate some values of these functions and print the results, along with your program.

E12.25. Continue in your file `recursive1.rb` to generate $\mathsf{lcm}(m,n)$ the least common multiple of $\mathsf{S}m$ and $\mathsf{S}n$ — that is, $(\mu y \leq \mathsf{S}m \times \mathsf{S}n)[y > 0 \wedge m|y \wedge n|y]$. For this you will need the characteristic function of $y > 0 \wedge m|y \wedge n|y$; and then one like $\mathsf{ch}_{\mathsf{R}}(m,n,v)$ corresponding to $(\exists y \leq v)[y > 0 \wedge m|y \wedge n|y]$. Then you will be able to find the function like $p(m,n,v)$ corresponding to $(\mu y \leq v)[y > 0 \wedge m|y \wedge n|y]$ and finally the lcm.

*E12.26. Functions $f_1(\vec{x}, y)$ and $f_2(\vec{x}, y)$ are defined by *simultaneous* (mutual) recursion just in case,

$f_1(\vec{x}, 0) = g_1(\vec{x})$

$f_2(\vec{x}, 0) = g_2(\vec{x})$

$f_1(\vec{x}, \mathsf{S}y) = h_1(\vec{x}, y, f_1(\vec{x}, y), f_2(\vec{x}, y))$

$f_2(\vec{x}, \mathsf{S}y) = h_2(\vec{x}, y, f_1(\vec{x}, y), f_2(\vec{x}, y))$

Show that f_1 and f_2 so defined are recursive. Hint: Let $F(\vec{x}, y) = \pi_0^{f_1(\vec{x},y)} \times \pi_1^{f_2(\vec{x},y)}$; then find $G(\vec{x})$ in terms of g_1 and g_2, and $H(\vec{x}, y, u)$ in terms of h_1 and h_2 so that $F(\vec{x}, 0) = G(\vec{x})$ and $F(\vec{x}, \mathsf{S}y) = H(\vec{x}, y, F(\vec{x}, y))$. So $F(\vec{x}, y)$ is recursive. Then $f_1(\vec{x}, y) = \mathsf{exp}(F(\vec{x}, y), 0)$ and $f_2(\vec{x}, y) = \mathsf{exp}(F(\vec{x}, y), 1)$; so f_1 and f_2 are recursive.

12.4.3 Arithmetization

Our aim in this section is to assign numbers to to expressions and sequences of expressions in $\mathscr{L}_{NT}$ and build a (primitive) recursive property $\mathsf{PRFQ}(m, n)$ which is true just in case m numbers a sequence of expressions that is a proof of the expression numbered by n. This requires a number of steps. In this part, we develop at lest the notion of a *sentential* proof which should be sufficient for the general idea. The next section develops details for the the full quantificational case.

Gödel numbers. We begin with a strategy familiar from 10.2.2 and 10.3.2 (to which you may find it helpful to refer), now adapted to $\mathscr{L}_{NT}$. The idea is to assign numbers to symbols and expressions of $\mathscr{L}_{NT}$. Then we shall be able to operate on the associated numbers by means of ordinary numerical functions. Insofar as the variable symbols in any quantificational language are countable, they are capable of being sorted into series, $x_1, x_2 \ldots$ Supposing that this is done, begin by assigning to each symbol α in $\mathscr{L}_{NT}$ an integer $g[\alpha]$ called its *Gödel Number*.

a.	$g[(] = 3$	f.	$g[\forall] = 13$
b.	$g[)] = 5$	g.	$g[\emptyset] = 15$
c.	$g[\sim] = 7$	h.	$g[S] = 17$
d.	$g[\rightarrow] = 9$	i.	$g[+] = 19$
e.	$g[=] = 11$	j.	$g[\times] = 21$

k. $g[x_i] = 23 + 2i$

So, for example, $g[x_5] = 23 + 2 \times 5 = 33$. Clearly each symbol gets a unique Gödel number, and Gödel numbers for individual symbols are odd positive integers.[8]

Now we are in a position to assign a Gödel number to each formula as follows: Where $\alpha_0, \alpha_1 \ldots \alpha_n$ are the symbols, in order from left to right, in some expression $\mathcal{Q}$,

$$g[\mathcal{Q}] = 2^{g[\alpha_0]} \times 3^{g[\alpha_1]} \times 5^{g[\alpha_2]} \times \ldots \times \pi_n{}^{g[\alpha_n]}$$

where 2, 3, 5… π_n are the first n prime numbers. So, for example, $g[x_0 \times x_5] = 2^{23} \times 3^{21} \times 5^{33}$. This is a big integer. But it is an integer, and different expressions get different Gödel numbers. Given a Gödel number, we can find the corresponding expression by finding its prime factorization; then if there are twenty three 2s in the factorization, the first symbol is x_0; if there are twenty one 3s, the second symbol is $\times$; and so forth. Notice that numbers for individual symbols are odd, where numbers for expressions are even.

Now consider a sequence of expressions, $\mathcal{Q}_0$, $\mathcal{Q}_1 \ldots \mathcal{Q}_n$ (as in an axiomatic derivation). These expressions have Gödel numbers g_0, $g_1 \ldots$, g_n. Then,

$$\pi_0^{g_0} \times \pi_1^{g_1}, \times \pi_2^{g_2} \times \ldots \times \pi_n^{g_n}$$

is the *super* Gödel number for the sequence $\mathcal{Q}_0$, $\mathcal{Q}_1 \ldots \mathcal{Q}_n$. Again, given a super Gödel number, we can find the corresponding expressions by finding its prime factorization; then, if there are g_0 2s, we can proceed to the prime factorization of g_0, to discover the symbols of the first expression; and so forth. Observe that super Gödel numbers are even, but are distinct from Gödel numbers for expressions, insofar as the exponent of 2 in the factorization of any expression is odd (the first element of any expression is a symbol and so has an odd number); and the exponent of 2 in the factorization of any super Gödel number is even (the first element of a sequence is an expression and so has an even number).

[8] There are many ways to do this, we pick just one.

Recall that $\mathsf{exp}(\mathsf{n},\mathsf{i})$ returns the exponent of π_{i} in the prime factorization of n. So for a Gödel number n, $\mathsf{exp}(\mathsf{n},\mathsf{i})$ returns the code of α_i; and for a super Gödel number n, $\mathsf{exp}(\mathsf{n},\mathsf{i})$ returns the code of $\mathcal{Q}_i$.

Where $\mathcal{P}$ is any expression, let $\ulcorner\mathcal{P}\urcorner$ be its Gödel number; and $\overline{\ulcorner\mathcal{P}\urcorner}$ the standard numeral for its Gödel number. In this case, say, $\ulcorner 0\urcorner = 2^{15}$ rather than 15 — for we take the number of the bracketed *expression*.

Concatenation. The function $\mathsf{cncat}(\mathsf{m},\mathsf{n})$ — ordinarily indicated $\mathsf{m}\star\mathsf{n}$, returns the Gödel number of the expression with Gödel number m followed by the expression with Gödel number n. So $\ulcorner x\times y\urcorner\star\ulcorner = z\urcorner = \ulcorner x\times y = z\urcorner$, for some numbered variables x, y and z. This function is (primitive) recursive. Recall that $\mathsf{len}(\mathsf{n})$ is recursive and returns the number of distinct prime factors of n. Set $\mathsf{m}\star\mathsf{n}$ to,

$$(\mu x \leq \mathsf{B}_{\mathsf{m},\mathsf{n}})[(\forall \mathsf{i} < \mathsf{len}(\mathsf{m}))\{\mathsf{exp}(\mathsf{x},\mathsf{i}) = \mathsf{exp}(\mathsf{m},\mathsf{i})\} \wedge (\forall \mathsf{i} < \mathsf{len}(\mathsf{n}))\{\mathsf{exp}(\mathsf{x},\mathsf{i}+\mathsf{len}(\mathsf{m})) = \mathsf{exp}(\mathsf{n},\mathsf{i})\}]$$

We search for the least number x such that exponents of initial primes in its factorization match the exponents of primes in m and exponents of primes later match eponents of primes in n. The bounded quantifiers take $\mathsf{i} < \mathsf{len}(\mathsf{m})$ and $\mathsf{i} < \mathsf{len}(\mathsf{n})$ insofar as len returns the number of primes, but $\mathsf{exp}(\mathsf{x},\mathsf{i})$ starts the list of primes at 0; so if $\mathsf{len}(\mathsf{m}) = 3$, its primes are π_0, π_1 and π_2. So the first $\mathsf{len}(\mathsf{m})$ exponents of x are the same as the exponents in m, and the next $\mathsf{len}(\mathsf{n})$ exponents of x are the same as the exponents in n.

To ensure that the function is recursive, we use the bounded least element quantifier as main operator, where $\mathsf{B}_{\mathsf{m},\mathsf{n}}$ is the bound under which we search for x. In this case it is sufficient to set

$$\mathsf{B}_{\mathsf{m},\mathsf{n}} = \left(\pi_{\mathsf{len}(\mathsf{m})+\mathsf{len}(\mathsf{n})}^{\mathsf{m}+\mathsf{n}}\right)^{\mathsf{len}(\mathsf{m})+\mathsf{len}(\mathsf{n})}$$

The idea is that all the primes in x will be $\leq \pi_{\mathsf{len}(\mathsf{m})+\mathsf{len}(\mathsf{n})}$. And any exponent in the factorization of m must be $\leq \mathsf{m}$ and any exponent for n must be $\leq \mathsf{n}$; so that $\mathsf{m}+\mathsf{n}$ is greater than any exponent in the factorization of x. So B results from multiplying a prime larger than any in x to a power greater than that of any in x together as many times as there are primes in x; so x must be smaller than B.

Observe that corresponding to association for multiplication $(\mathsf{m}\star\mathsf{n})\star\mathsf{o} = \mathsf{m}\star(\mathsf{n}\star\mathsf{o})$; so we often drop parentheses for the concatenation operation.

Terms and Atomics. $\textsc{term}(\mathsf{n})$ is true iff n is the Gödel number of a term. Think of the trees on which we show that an expression is a term. Put formally, for any term

t_n, there is a *term sequence* $t_0, t_1 \ldots t_n$ such that each expression is either,

a. $\emptyset$

b. a variable

c. St_j where t_j occurs earlier in the sequence

d. $+t_i t_j$ where t_i and t_j occur earlier in the sequence

e. $\times t_i t_j$ where t_i and t_j occur earlier in the sequence

where we represent terms in unabbreviated form. A term is the last element of such a sequence. Let us try to say this.

First, $\mathsf{VAR}(\mathsf{n})$ is true just in case n is the Gödel number of a variable — conceived as an expression, rather than a symbol. Then VAR is (primitive) recursive. Set,

$$\mathsf{VAR}(\mathsf{n}) =_{\text{def}} (\exists \mathsf{x} \leq \mathsf{n})(\mathsf{n} = 2^{23+2\mathsf{x}})$$

If there is such an x, then n must be the Gödel number of a variable. And it is clear that this x is less than n itself. So the result is recursive.

Now $\mathsf{TERMSEQ}(\mathsf{m}, \mathsf{n})$ is true when m is the super Gödel number of a sequence of expressions whose last member has Gödel number n. For $\mathsf{TERMSEQ}(\mathsf{m}, \mathsf{n})$ set,

$$\begin{aligned}
&\mathsf{exp}(\mathsf{m}, \mathsf{len}(\mathsf{m}) \dot{-} 1) = \mathsf{n} \wedge \mathsf{m} > 1 \wedge (\forall \mathsf{k} < \mathsf{len}(\mathsf{m}))\{\\
&\mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner\emptyset\urcorner \vee \mathsf{VAR}(\mathsf{exp}(\mathsf{m}, \mathsf{k})) \vee\\
&(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner S\urcorner \star \mathsf{exp}(\mathsf{m}, \mathsf{j})] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner +\urcorner \star \mathsf{exp}(\mathsf{m}, \mathsf{i}) \star \mathsf{exp}(\mathsf{m}, \mathsf{j})] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner \times\urcorner \star \mathsf{exp}(\mathsf{m}, \mathsf{i}) \star \mathsf{exp}(\mathsf{m}, \mathsf{j})]\}
\end{aligned}$$

Recall that $\mathsf{len}(\mathsf{m})$ returns the number of primes in the prime factorization of m; so supposing that m is other than zero or one, $\mathsf{len}(\mathsf{m}) \geq 1$ and if there is one prime it is π_0, if there are two primes they are π_0 and π_1, etc. So the last member of the sequence has Gödel number n and any member of the sequence is a constant or a variable, or made up in the usual way by prior members.

Then set $\mathsf{TERM}(\mathsf{n})$ as follows,

$$\mathsf{TERM}(\mathsf{n}) =_{\text{def}} (\exists \mathsf{x} \leq \mathsf{B}_\mathsf{n})\mathsf{TERMSEQ}(\mathsf{x}, \mathsf{n})$$

If some x numbers a term sequence for n, then n is a term. In this case, the Gödel numbers of all prior members in the sequence must be less than n. Further, the number of members in the sequence is the same as the number of variables and

constants together with the number of function symbols in the term (one member for each variable and constant, and another corresponding to each function symbol); so the number of members in the sequence is the same as $\mathsf{len(n)}$; so all the primes in the sequence are $< \pi_{\mathsf{len(n)}}$. So multiply $\pi^{\mathsf{n}}_{\mathsf{len(n)}}$ together $\mathsf{len(n)}$ times and set $\mathsf{B_n} = \left(\pi^{\mathsf{n}}_{\mathsf{len(n)}}\right)^{\mathsf{len(n)}}$. We take a prime $\pi_{\mathsf{len(n)}}$ greater than all the primes in the sequence, to a power n greater than all the powers in the sequence, and multiply it together as many times as there are members of the sequence. The result must be greater than x, the number of the term sequence.

Finally $\textsc{atom}(\mathsf{n})$ is true iff n is the number of an atomic formula. The only atomic formulas of $\mathcal{L}_{\mathrm{NT}}$ are of the form $=t_1 t_2$. So it is sufficient to set,

$$\textsc{atom}(\mathsf{n}) =_{\mathrm{def}} (\exists \mathsf{x} \leq \mathsf{n})(\exists \mathsf{y} \leq \mathsf{n})[\textsc{term}(\mathsf{x}) \wedge \textsc{term}(\mathsf{y}) \wedge \mathsf{n} = \ulcorner = \urcorner \star \mathsf{x} \star \mathsf{y}]$$

Clearly the numbers of t_1 and t_2 are $\leq \mathsf{n}$ itself.

Formulas. $\textsc{wff}(\mathsf{n})$ is to be true iff n is the number of a (well-formed) formula. Again, think of the tree by which a formula is formed. There is a sequence of which each member is,

a. an atomic

b. $\sim\mathcal{P}$ for some previous member of the sequence $\mathcal{P}$

c. $(\mathcal{P} \rightarrow \mathcal{Q})$ for previous members of the sequence $\mathcal{P}$ and $\mathcal{Q}$

d. $\forall x \mathcal{P}$ for some previous member of the sequence $\mathcal{P}$ and variable x

So, on the model of what has gone before, we let $\textsc{formseq}(\mathsf{m}, \mathsf{n})$ be true when m is the super Gödel number of a sequence of formulas whose last member has Gödel number n. For $\textsc{formseq}(\mathsf{m}, \mathsf{n})$ set,

$$\begin{aligned}
&\mathsf{exp(m, len(m)} \dot{-} 1) = \mathsf{n} \wedge \mathsf{m} > 1 \wedge (\forall \mathsf{k} < \mathsf{len(m)})\{\\
&\textsc{atomic}(\mathsf{exp(m, k)}) \vee\\
&(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp(m, k)} = \ulcorner \sim \urcorner \star \mathsf{exp(m, j)}] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp(m, k)} = \ulcorner (\urcorner \star \mathsf{exp(m, i)} \star \ulcorner \rightarrow \urcorner \star \mathsf{exp(m, j)} \star \ulcorner) \urcorner] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{n})[\textsc{var}(\mathsf{j}) \wedge \mathsf{exp(m, k)} = \ulcorner \forall \urcorner \star \mathsf{j} \star \mathsf{exp(m, i)}]\}
\end{aligned}$$

So a formula is the last member of a sequence each member of which is an atomic, or formed from previous members in the usual way. Clearly the number of a variable in an expression with number n is itself $\leq \mathsf{n}$. Then,

$$\text{WFF}(n) =_{def} (\exists x \leq B_n)(\text{FORMSEQ}(x, n))$$

An expression is a formula iff there is a formula sequence of which it is the last member. Again, the Gödel numbers of all the prior formulas in the sequence must be $\leq n$. And there are as many members of the sequence as there are atomics and operator symbols in the formula numbered n. So all the primes are $\leq \pi_{len(n)}$; so multiply So multiply $\pi^n_{len(n)}$ together $len(n)$ times and set $B_n = \left(\pi^n_{len(n)}\right)^{len(n)}$.

Sentential Proof. $\text{SENTPRF}(m, n)$ is to be true iff m is the super Gödel number of a sequence of formulas that is a (sentential) proof of the formula with Gödel number n. We revert to the relatively simple axiomatic system of chapter 3. So, for example, A1 is of the sort, $(\mathcal{P} \rightarrow (\mathcal{Q} \rightarrow \mathcal{P}))$, and the only rule is MP. For the sentential case we need, $\text{SENTAXIOM}(n)$ true when n is the number of an axiom. For this,

$\text{AXIOM1}(n) =_{def}$ $(\exists x \leq n)(\exists y \leq n)[\text{WFF}(x) \wedge \text{WFF}(y) \wedge n = \ulcorner(\urcorner \star x \star \ulcorner\rightarrow\urcorner \star \ulcorner(\urcorner \star y \star \ulcorner\rightarrow\urcorner \star x \star \ulcorner))\urcorner]$

$\text{AXIOM2}(n) =_{def}$ Homework.

$\text{AXIOM3}(n) =_{def}$ Homework.

Then,

$$\text{SENTAXIOM}(n) =_{def} \text{AXIOM1}(n) \vee \text{AXIOM2}(n) \vee \text{AXIOM3}(n)$$

In the next section, we will add all the logical axioms plus the axioms for Q. But this is all that is required for proofs of theorems of sentential logic.

Now $cnd(n, o) = m$ when $n = \ulcorner\mathcal{P}\urcorner$, $o = \ulcorner\mathcal{Q}\urcorner$ and $m = \ulcorner(\mathcal{P} \rightarrow \mathcal{Q})\urcorner$. And $\text{MP}(m, n, o)$ is true when the formula with Gödel number o follows from ones with numbers m and n.

$$cnd(n, o) = \ulcorner(\urcorner \star n \star \ulcorner\rightarrow\urcorner \star o \star \ulcorner)\urcorner$$

$$\text{MP}(m, n, o) =_{def} cnd(n, o) = m$$

So m numbers the conditional, n its antecedent, and o the consequent.

And $\text{SENTPRF}(m, n)$ when m is the super Gödel number of a sequence that is a proof whose last member has Gödel number n. This works like TERMSEQ and FORMSEQ. For SENTPRF set,

$exp(m, len(m) \dot{-} 1) = n \wedge m > 1 \wedge (\forall k < len(m))\{$

$\text{SENTAXIOM}(exp(m, k)) \vee$

$(\exists i < k)(\exists j < k)\textsc{mp}(\mathsf{exp}(m, i), \mathsf{exp}(m, j), \mathsf{exp}(m, k))\}$

So every formula is either an axiom, or follows from previous members by MP. It is a significant matter to have shown that there is such a function! Again, in the next section, we will extend this notion to include the rule Gen.

This construction for SENTPRF exhibits the essential steps that are required for the parallel relation $\textsc{prfq}(m, n)$ for theorems of Q. That discussion is taken up in the following section, and adds considerable detail. It is not clear that the detail is required for understanding results to follow — though of course, to the extent that those results rely on the recursive PRFQ relation, the detail underlies *proof* of the results!

E12.27. Find Gödel numbers for each of the following. Treat the first as an expression, rather than as simple symbol; the last is a sequence of expressions. For the latter two, you need not do the calculation!

$$x_2 \qquad x_0 = x_1 \qquad x_0 = x_1, \emptyset = x_0, \emptyset = x_1$$

E12.28. Complete the cases for $\textsc{axiom2}(n)$ and $\textsc{axiom3}(n)$.

E12.29. In chapter 8 we define the notion of a *normal* sentential form (p. 389). Using ATOM from above, define a recursive relation $\textsc{norm}(n)$ for $\mathscr{L}_{NT}$. Hint: You will need a formula sequence to do this.

12.4.4 Completing the Construction

Quantifier rules for derivations include axioms like (A4), $(\forall v \mathcal{P} \rightarrow \mathcal{P}^v_s)$ where term s is free for variable v in $\mathcal{P}$. This is easy enough to apply in practice. But it takes some work to represent. We tackle the problem piece-by-piece.

Substitution in terms. Say $t = \ulcorner t \urcorner$, $v = \ulcorner v \urcorner$, and $s = \ulcorner s \urcorner$ for some terms s, t, and variable v. Then $\textsc{termsub}(t, v, s, u)$ is true when u is the Gödel number of t^v_s. For this, we begin with a term sequence (with Gödel number m) for t, and consider a parallel sequence, not necessarily a term sequence (with Gödel number n), that includes modified versions of the terms in the sequence with Gödel number m. For $\textsc{termsub}(t, v, s, u)$ set,

$$
\begin{aligned}
&(\exists \mathsf{m} \leq \mathsf{X})(\exists \mathsf{n} \leq \mathsf{Y})(\textsc{termseq}(\mathsf{m},\mathsf{t}) \wedge \mathsf{exp}(\mathsf{n},\mathsf{len}(\mathsf{n}) \dot{-} 1) = \mathsf{u}) \wedge \mathsf{n} > 1 \wedge (\forall \mathsf{k} < \mathsf{len}(\mathsf{n}))\{\\
&[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \ulcorner \emptyset \urcorner \wedge \mathsf{exp}(\mathsf{n},\mathsf{k}) = \ulcorner \emptyset \urcorner] \vee\\
&[\textsc{var}(\mathsf{exp}(\mathsf{m},\mathsf{k})) \wedge \mathsf{exp}(\mathsf{m},\mathsf{k}) \neq \mathsf{v} \wedge \mathsf{exp}(\mathsf{n},\mathsf{k}) = \mathsf{exp}(\mathsf{m},\mathsf{k})] \vee\\
&[\textsc{var}(\mathsf{exp}(\mathsf{m},\mathsf{k})) \wedge \mathsf{exp}(\mathsf{m},\mathsf{k}) = \mathsf{v} \wedge \mathsf{exp}(\mathsf{n},\mathsf{k}) = \mathsf{s}] \vee\\
&(\exists \mathsf{i} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \ulcorner S \urcorner \star \mathsf{exp}(\mathsf{m},\mathsf{i}) \wedge \mathsf{exp}(\mathsf{n},\mathsf{k}) = \ulcorner S \urcorner \star \mathsf{exp}(\mathsf{n},\mathsf{i})] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \ulcorner + \urcorner \star \mathsf{exp}(\mathsf{m},\mathsf{i}) \star \mathsf{exp}(\mathsf{m},\mathsf{j}) \wedge \mathsf{exp}(\mathsf{n},\mathsf{k}) = \ulcorner + \urcorner \star \mathsf{exp}(\mathsf{n},\mathsf{i}) \star \mathsf{exp}(\mathsf{n},\mathsf{j})] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m},\mathsf{k}) = \ulcorner \times \urcorner \star \mathsf{exp}(\mathsf{m},\mathsf{i}) \star \mathsf{exp}(\mathsf{m},\mathsf{j}) \wedge \mathsf{exp}(\mathsf{n},\mathsf{k}) = \ulcorner \times \urcorner \star \mathsf{exp}(\mathsf{n},\mathsf{i}) \star \mathsf{exp}(\mathsf{n},\mathsf{j})]\}
\end{aligned}
$$

So the sequence for t^{v}_{s} (numbered by n) is like one of our "unabbreviating trees" from chapter 2. In any place where the sequence for t (numbered by m) numbers $\emptyset$, the sequence for t^{v}_{s} numbers $\emptyset$. Where the sequence for t numbers a variable other than v, the sequence for t^{v}_{s} numbers the same variable. But where the sequence for t numbers variable v, the sequence for t^{v}_{s} numbers s. Then later parts are built out of prior in parallel. The second sequence may not itself be a *term* sequence, insofar as it need not include all the antecedents to s (just as an unabbreviating tree would not include all the parts of a resultant term or formula).

In this case, reasoning as for WFF, the Gödel numbers in the sequence with number m must be less than t and numbers in the sequence with number n must be less than u. And primes in the sequence range up to $\pi_{\mathsf{len(t)}}$. So it is sufficient to set $\mathsf{X} = \left(\pi^{\mathsf{t}}_{\mathsf{len(t)}}\right)^{\mathsf{len(t)}}$ and $\mathsf{Y} = \left(\pi^{\mathsf{u}}_{\mathsf{len(t)}}\right)^{\mathsf{len(t)}}$.

Substitution in atomics. Say $\mathsf{p} = \ulcorner \mathcal{P} \urcorner$, $\mathsf{v} = \ulcorner v \urcorner$, and $\mathsf{s} = \ulcorner s \urcorner$ for some atomic formula $\mathcal{P}$, variable v and term s. Then ATOMSUB$(\mathsf{p},\mathsf{v},\mathsf{s},\mathsf{u})$ is true when u is the Gödel number of $\mathcal{P}^{v}_{s}$. The condition is straightforward given TERMSUB. For ATOMSUB$(\mathsf{p},\mathsf{v},\mathsf{s},\mathsf{u})$,

$$(\exists \mathsf{i} \leq \mathsf{p})(\exists \mathsf{j} \leq \mathsf{p})(\exists \mathsf{i}' \leq \mathsf{u})(\exists \mathsf{j}' \leq \mathsf{u})[\textsc{term}(\mathsf{i}) \wedge \textsc{term}(\mathsf{j}) \wedge \mathsf{p} = \ulcorner = \urcorner \star \mathsf{i} \star \mathsf{j} \wedge \textsc{termsub}(\mathsf{i},\mathsf{v},\mathsf{s},\mathsf{i}') \wedge \textsc{termsub}(\mathsf{j},\mathsf{v},\mathsf{s},\mathsf{j}') \wedge \mathsf{u} = \ulcorner = \urcorner \star \mathsf{i}' \star \mathsf{j}']$$

$\mathcal{P}^{v}_{s}$ simply substitutes into the terms on either side of the equal sign.

Substitution into formulas. In the general case, $\mathcal{P}^{v}_{s}$ is complicated insofar as s replaces only *free* instances of v. Again, we build a parallel sequence with number n. No replacements are carried forward in subformulas beginning with a quantifier binding instances of variable v. Where $\mathsf{p} = \ulcorner \mathcal{P} \urcorner$, $\mathsf{v} = \ulcorner v \urcorner$, and $\mathsf{s} = \ulcorner s \urcorner$ for an arbitrary formula $\mathcal{P}$, variable v and term s, FORMSUB$(\mathsf{p},\mathsf{v},\mathsf{s},\mathsf{u})$ is true when u is the Gödel number of $\mathcal{P}^{v}_{s}$. For this set,

$(\exists m \leq X)(\exists n \leq Y)(\text{FORMSEQ}(m,p) \wedge exp(n, len(n) \dot{-} 1) = u) \wedge n > 1 \wedge (\forall k < len(n))\{$
$[\text{ATOM}(exp(m,k)) \wedge \text{ATOMSUB}(exp(m,k), v, s, exp(n,k))] \vee$
$(\exists i < k)[exp(m,k) = \ulcorner \sim \urcorner \star exp(m,i) \wedge exp(n,k) = \ulcorner \sim \urcorner \star exp(n,i)] \vee$
$(\exists i < k)(\exists j < k)[exp(m,k) = \ulcorner (\urcorner \star exp(m,i) \star \ulcorner \rightarrow \urcorner \star exp(m,j) \star \ulcorner) \urcorner \wedge exp(n,k) = \ulcorner (\urcorner \star exp(n,i) \star \ulcorner \rightarrow \urcorner \star exp(n,j) \star \ulcorner) \urcorner] \vee$
$(\exists i < k)(\exists j < p)[\text{VAR}(j) \wedge j \neq v \wedge exp(m,k) = \ulcorner \forall \urcorner \star j \star exp(m,i) \wedge exp(n,k) = \ulcorner \forall \urcorner \star j \star exp(n,i)] \vee$
$(\exists i < k)(\exists j < p)[\text{VAR}(j) \wedge j = v \wedge exp(m,k) = \ulcorner \forall \urcorner \star j \star exp(m,i) \wedge exp(n,k) = exp(m,k)]\}$

So substitutions are made in atomics, and carried forward in the parallel sequence — so long as no quantifier binds variable v, at which stage, the sequence reverts to the form without substitution. Again, set $X = \left(\pi^{p}_{len(p)}\right)^{len(p)}$ and $Y = \left(\pi^{u}_{len(p)}\right)^{len(p)}$.

Given $\text{FORMSUB}(p, v, s, u)$, there is a corresponding function $formusb(p, v, s) = (\mu u \leq Z)(\text{FORMSUB}(p, v, s, u)$. In this case, the number of symbols in $\mathcal{P}^{v}_{s}$ is sure to be no greater than the number of symbols in $\mathcal{P}$ times the number of symbols in s. And the Gödel number of each symbol is no greater than the maximum of p and s and so $p + s$. So it is sufficient to set $Z = \left(\pi^{p+s}_{len(p)\times len(s)}\right)^{len(p)\times len(s)}$. Again, we take a prime at least great as that of any symbol, to a power greater than that of any exponent, and multiply it as many times as there are symbols.

Free and bound variables. $\text{FREE}(p, v)$ is true when v is the Gödel number of a variable that is free in a term or formula with Gödel number p. For a given variable x_i initially assigned number $23 + 2i$, $\ulcorner x_i \urcorner = 2^{23+2i}$; and $\ulcorner x_i \urcorner^2 = 2^{23+2i+2}$ is the number of the next variable. In particular then, for v the number of a variable, v^2 numbers a different variable. The idea is that if there is some change in an expression upon substitution of a variable different from v, then v must have been free in the original expression. For terms and formulas respectively,

$\text{FREEt}(t, v) =_{def} \sim\text{TERMSUB}(t, v, v^2, t)$

$\text{FREEf}(p, v) =_{def} \sim\text{FORMSUB}(p, v, v^2, p)$

So v is free if the result upon substitution is other than the original expression.

Given $\text{FREEf}(p, v)$, it is a simple matter to specify $\text{SENT}(n)$ true when n numbers a sentence.

$$\text{SENT}(n) =_{def} \text{WFF}(n) \wedge (\forall x < n)[\text{VAR}(x) \rightarrow \sim\text{FREEf}(n, x)]$$

So n numbers a sentence if it numbers a formula and nothing is a number of a variable free in the formula numbered by n.

Finally, suppose $s = \ulcorner s \urcorner$ and $v = \ulcorner v \urcorner$; then $\text{FREEFOR}(s, v, u)$ is true iff s is free for v in the formula numbered by u. For this, we set up a modified formula

sequence, that identifies just "admissable" subformulas — ones where $\mathscr{s}$ is free for v in the formula numbered by u. For $\textsc{ffseq}(\mathsf{m}, \mathsf{s}, \mathsf{v}, \mathsf{u})$ set,

$$\begin{aligned}
&\mathsf{exp}(\mathsf{m}, \mathsf{len}(\mathsf{m}) \dot{-} 1) = \mathsf{u} \wedge \mathsf{m} > 1 \wedge (\forall \mathsf{k} < \mathsf{len}(\mathsf{m}))\{\\
&\textsc{atomic}(\mathsf{exp}(\mathsf{m}, \mathsf{k})) \vee\\
&(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner\sim\urcorner \star \mathsf{exp}(\mathsf{m}, \mathsf{j})] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{k})[\mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner(\urcorner \star \mathsf{exp}(\mathsf{m}, \mathsf{i}) \star \ulcorner\rightarrow\urcorner \star \mathsf{exp}(\mathsf{m}, \mathsf{j}) \star \ulcorner)\urcorner] \vee\\
&(\exists \mathsf{j} < \mathsf{u})[\textsc{wff}(\mathsf{j}) \wedge \mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner\forall\urcorner \star \mathsf{v} \star \mathsf{j}] \vee\\
&(\exists \mathsf{i} < \mathsf{k})(\exists \mathsf{j} < \mathsf{u})[\textsc{var}(\mathsf{j}) \wedge \mathsf{j} \neq \mathsf{v} \wedge (\textsc{freet}(\mathsf{s}, \mathsf{j}) \rightarrow \sim\textsc{freef}(\mathsf{exp}(\mathsf{m}, \mathsf{i}), \mathsf{v})) \wedge \mathsf{exp}(\mathsf{m}, \mathsf{k}) = \ulcorner\forall\urcorner \star \mathsf{j} \star \mathsf{exp}(\mathsf{m}, \mathsf{i})]\}
\end{aligned}$$

If the main operator of a subformula $\mathscr{Q}$ binds variable v, then no variables in $\mathscr{s}$ are bound upon substitution, because there are no substitutions — as only free instances of v are replaced. Observe that this $\mathscr{Q}$ need not appear earlier in the sequence, as any formula with the v quantifier satisfies the condition. Alternatively, if the main operator binds a different variable, we require that either the variable is not free in $\mathscr{s}$ or v is not free in $\mathscr{Q}$, else variables in $\mathscr{s}$ become bound upon substitution. Given this,

$$\textsc{freefor}(\mathsf{s}, \mathsf{v}, \mathsf{u}) =_{\text{def}} (\exists \mathsf{x} < \mathsf{B_u})\textsc{ffseq}(\mathsf{x}, \mathsf{s}, \mathsf{v}, \mathsf{u})$$

In this case, every member of the sequence for FFSEQ is a member of the FORMSEQ for u so $\mathsf{B_u}$ may be set as before.

Proofs. After all this work, we are finally ready for axiom 4. $\textsc{axiom4}(\mathsf{n})$ obtains when n is the Gödel number of an instance of A4.

$$(\exists \mathsf{p} \leq \mathsf{n})(\exists \mathsf{v} \leq \mathsf{n})(\exists \mathsf{s} \leq \mathsf{n})[\textsc{wff}(\mathsf{p}) \wedge \textsc{var}(\mathsf{v}) \wedge \textsc{term}(\mathsf{s}) \wedge \textsc{freefor}(\mathsf{s}, \mathsf{v}, \mathsf{p}) \wedge \mathsf{n} = \mathsf{cnd}(\ulcorner\forall\urcorner \star \mathsf{v} \star \mathsf{p}, \mathsf{formsub}(\mathsf{p}, \mathsf{v}, \mathsf{s}))]$$

So there is a formula $\mathscr{P}$, variable v and term $\mathscr{s}$ where $\mathscr{s}$ is free for v in $\mathscr{P}$; and the axiom is of the form, $(\forall v \mathscr{P} \rightarrow \mathscr{P}^{v}_{\mathscr{s}})$.

$\textsc{gen}(\mathsf{m}, \mathsf{n})$ holds when n is the Gödel number of a formula that follows from a formula with Gödel number m. Hint: you will need to assert the existence of numbers for formulas $\mathscr{P}$, $\mathscr{Q}$ and variable v, where v is not free in $\mathscr{P}$. Then simply require that m numbers a formula of the sort $(\mathscr{P} \rightarrow \mathscr{Q})$ and n one of the sort $(\mathscr{P} \rightarrow \forall v \mathscr{Q})$.

After what we have done, axioms for equality and Robinson Arithmetic are not hard. A few are worked as examples.

$$\textsc{axiom5}(\mathsf{n}) =_{\text{def}} (\exists \mathsf{v} \leq \mathsf{n})[\textsc{var}(\mathsf{v}) \wedge \mathsf{n} = \mathsf{v} \star \ulcorner=\urcorner \star \mathsf{v}]$$

For "simplicity" I drop the unabbreviated style of the original formulas.

Axiom six is of the sort, $(x_i = y) \to (h^n x_1 \ldots x_i \ldots x_n = h^n x_1 \ldots y \ldots x_n)$ for relation symbol h and variables $x_1 \ldots x_n$ and y. In $\mathcal{L}_{NT}$ the function symbol is S, $+$ or $\times$. Because just a single replacement is made, we do not want to use TERMSUB. However, we are in a position simply to list all the combinations in which one variable is replaced. So, for AXIOM6(n),

$$
\begin{aligned}
&(\exists \mathsf{s} < \mathsf{n})(\exists \mathsf{t} < \mathsf{n})(\exists \mathsf{x} < \mathsf{n})(\exists \mathsf{y} < \mathsf{n})\{\text{VAR}(\mathsf{x}) \land \text{VAR}(\mathsf{y}) \land (\\
&[\mathsf{s} = \ulcorner S \urcorner \star \mathsf{x} \land \mathsf{t} = \ulcorner S \urcorner \star \mathsf{y}] \lor\\
&(\exists \mathsf{z} < \mathsf{n})[\text{VAR}(\mathsf{z}) \land ((\mathsf{s} = \ulcorner + \urcorner \star \mathsf{x} \star \mathsf{z} \land \mathsf{t} = \ulcorner + \urcorner \star \mathsf{y} \star \mathsf{z}) \lor (\mathsf{s} = \ulcorner + \urcorner \star \mathsf{z} \star \mathsf{x} \land \mathsf{t} = \ulcorner + \urcorner \star \mathsf{z} \star \mathsf{y}))] \lor\\
&(\exists \mathsf{z} < \mathsf{n})[\text{VAR}(\mathsf{z}) \land ((\mathsf{s} = \ulcorner \times \urcorner \star \mathsf{x} \star \mathsf{z} \land \mathsf{t} = \ulcorner \times \urcorner \star \mathsf{y} \star \mathsf{z}) \lor (\mathsf{s} = \ulcorner \times \urcorner \star \mathsf{z} \star \mathsf{x} \land \mathsf{t} = \ulcorner \times \urcorner \star \mathsf{z} \star \mathsf{y}))]\ ) \land\\
&\mathsf{n} = \ulcorner (= \urcorner \star \mathsf{x} \star \mathsf{y} \star \ulcorner \to\ = \urcorner \star \mathsf{s} \star \mathsf{t} \star \ulcorner) \urcorner\}
\end{aligned}
$$

So there is a term s and a term t which replaces one instance of x in s with y. Then the axiom is of the sort $=xy \to =st$.

Axiom seven is similar. It is stated in terms of atomics of the sort $\mathcal{R}^n x_1 \ldots x_n$ for relation symbol $\mathcal{R}$ and variables $x_1 \ldots x_n$. In $\mathcal{L}_{NT}$ the relation symbol is the equals sign, so these atomics are of the form, $x = y$. Again, because just a single replacement is made, we do not want to use FORMSUB. However, we may proceed by analogy with AXIOM6. This is left as an exercise.

The axioms of Q are particular sentences. So, for example, axiom Q2 is of the sort, $(Sx = Sy) \to (x = y)$. Let x and y be x_0 and x_1 respectively. Then,

$$\text{AXIOMQ2}(\mathsf{n}) =_{\text{def}} \mathsf{n} = \ulcorner (Sx = Sy) \to (x = y) \urcorner$$

For "ease of reading," I do not reduce it to unabbreviated form. Other axioms of Q may be treated in the same way.

And now it is straightforward to produce generalized versions of AXIOM(n) and PRFQ(m, n). For the latter, it will be convenient to have a relation ICON(m, n, o) true when the formula with Gödel number o is an *immediate consequence* of ones numbered m and n

$$\text{ICON}(\mathsf{m}, \mathsf{n}, \mathsf{o}) =_{\text{def}} \text{MP}(\mathsf{m}, \mathsf{n}, \mathsf{o}) \lor (\mathsf{m} = \mathsf{n} \land \text{GEN}(\mathsf{n}, \mathsf{o}))$$

It is a significant matter to have found these functions. Now we put them to work.

E12.30. Complete the construction with recursive relations for GEN(m, n), AXIOM7(n), the remaining axioms for Robinson arithmetic, and then AXIOM(n) and PRFQ(m, n).

First Results of Chapter 12

T12.1 For an interpretation with the required variable-free terms: (a) If $\mathscr{R}$ is a relation symbol and R is a relation, and $\mathsf{I}[\mathscr{R}] = \mathsf{R}(\mathsf{x}_1 \ldots \mathsf{x}_n)$, then $\mathsf{R}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is expressed by $\mathscr{R}x_1 \ldots x_n$. And (b) if $\hbar$ is a function symbol and h is a function and $\mathsf{I}[\hbar] = \mathsf{h}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ then $\mathsf{h}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is expressed by $\hbar x_1 \ldots x_n = v$.

T12.2 Suppose function $\mathsf{f}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is expressed by formula $\mathscr{F}(x_1 \ldots x_n, y)$; then if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$, $\mathsf{I}[\sim\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})] = \mathsf{T}$.

T12.3 On the standard interpretation N of $\mathscr{L}_{NT}$, each recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is expressed by some formula $\mathscr{F}(\vec{x}, v)$. Corollary: On the standard interpretation N of $\mathscr{L}_{NT}$, each recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ is expressed by some formula $\mathscr{R}(\vec{x})$.

T12.4 If T includes Q and function $\mathsf{f}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is captured by formula $\mathscr{F}(x_1 \ldots x_n, y)$ so that conditions (f.i) and (f.ii) hold, then if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_n\rangle, \mathsf{a}\rangle \notin \mathsf{f}$ then $T \vdash \sim\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_n, \overline{\mathsf{a}})$.

T12.5 On the standard interpretation N for $\mathscr{L}_{NT}$, (i) $\mathsf{N_d}[s \leq t] = \mathsf{S}$ iff $\mathsf{N_d}[s] \leq \mathsf{N_d}[t]$, and (ii) $\mathsf{N_d}[s < t] = \mathsf{S}$ iff $\mathsf{N_d}[s] < \mathsf{N_d}[t]$.

T12.6 On the standard interpretation N for $\mathscr{L}_{NT}$, (i) $\mathsf{N_d}[(\forall x \leq t)\mathscr{P}] = \mathsf{S}$ iff for every $\mathsf{m} \leq \mathsf{N_d}[t]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathscr{P}] = \mathsf{S}$ and (ii), $\mathsf{N_d}[(\forall x < t)\mathscr{P}] = \mathsf{S}$ iff for every $\mathsf{m} < \mathsf{N_d}[t]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathscr{P}] = \mathsf{S}$.

T12.7 On the standard interpretation N for $\mathscr{L}_{NT}$, (i) $\mathsf{N_d}[(\exists x \leq t)\mathscr{P}] = \mathsf{S}$ iff for some $\mathsf{m} \leq \mathsf{N_d}[t]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathscr{P}] = \mathsf{S}$ and (ii), $\mathsf{N_d}[(\exists x < t)\mathscr{P}] = \mathsf{S}$ iff for some $\mathsf{m} < \mathsf{N_d}[t]$, $\mathsf{N}_{\mathsf{d}(x|\mathsf{m})}[\mathscr{P}] = \mathsf{S}$.

T12.8 For any Δ_0 sentence $\mathscr{P}$, if $\mathsf{N}[\mathscr{P}] = \mathsf{T}$, then $\mathrm{Q} \vdash_{ND} \mathscr{P}$, and if $\mathsf{N}[\mathscr{P}] \neq \mathsf{T}$, then $\mathrm{Q} \vdash_{ND} \sim\mathscr{P}$.

T12.9 For any Σ_1 sentence $\mathscr{P}$ if $\mathsf{N}[\mathscr{P}] = \mathsf{T}$, then $\mathrm{Q} \vdash_{ND} \mathscr{P}$.

T12.10 The original formula by which any recursive function is expressed is Σ_1.

T12.11 On the standard interpretation N for $\mathscr{L}_{NT}$, any recursive formula is captured by the original formula by which it is expressed in $\mathrm{Q_s}$.

T12.12 Suppose $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ results by recursion from functions $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ where $\mathsf{g}(\vec{\mathsf{x}})$ is captured by some $\mathscr{G}(\vec{x}, z)$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ by $\mathscr{H}(\vec{x}, y, u, z)$. Then for the original expression $\mathscr{F}(\vec{x}, y, z)$ of $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$, if $\langle\langle \mathsf{m}_1 \ldots \mathsf{m}_b, \mathsf{n}\rangle, \mathsf{a}\rangle \in \mathsf{f}$, $\mathrm{Q_s} \vdash \forall w[\mathscr{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_b, \overline{\mathsf{n}}, w) \rightarrow w = \overline{\mathsf{a}}]$.

T12.13 If a function $\mathsf{f}(\mathsf{x}_1 \ldots \mathsf{x}_n)$ is expressed by a Δ_0 formula $\mathscr{F}(x_1 \ldots x_n, y)$, then there is a Δ_0 formula $\mathscr{F}'$ that captures f in Q.

T12.14 For $\mathscr{F}'(\vec{x}, y) =_{\text{def}} \mathscr{F}(\vec{x}, y) \wedge (\forall z \leq y)[\mathscr{F}(\vec{x}, z) \rightarrow z = y]$, and for any n, $\mathrm{Q} \vdash \forall \vec{x} \forall y[(\mathscr{F}'(\vec{x}, \overline{\mathsf{n}}) \wedge \mathscr{F}'(\vec{x}, y)) \rightarrow y = \overline{\mathsf{n}}]$.

T12.15 If $\mathscr{F}(\vec{x}, y)$ expresses $\mathsf{f}(\vec{\mathsf{x}})$, then $\mathscr{F}'(\vec{x}, y) = \mathscr{F}(\vec{x}, y) \wedge (\forall z < y)[\mathscr{F}(\vec{x}, z) \rightarrow z = y]$ expresses $\mathsf{f}(\vec{\mathsf{x}})$.

T12.16 Any recursive function is captured by a Σ_1 formula in Q. Corollary: Any recursive relation is captured by a Σ_1 formula in Q.

12.5 Essential Results

In this section, we develop some first fruits of our labor. We shall need some initial theorems, important in their own right. With these theorems in hand, our results follow in short order. The results are developed and extended in later chapters. But it is worth putting them on the table at the start. (And some results at this stage provide a fitting cap to our labors.) We have expended a great deal of energy showing that, under appropriate conditons, recursive functions can be expressed and captured, and that there are recursive functions and relations including PRFQ. Now we put these results to work.

12.5.1 Preliminary Theorems

A couple of definitions: If f is a function from (an initial segment of) $\mathbb{N}$ onto some set — so that the objects in the set are $\mathsf{f}(0)$, $\mathsf{f}(1)\ldots$ say f *enumerates* the members of the set. A set is *recursively* enumerable if there is a recursive function that enumerates it. Also, say T is a *recursively axiomatized* formal theory if there is a recursive relation $\text{PRFT}(\mathsf{m}, \mathsf{n})$ which holds just in case m is the super Gödel number of a proof in T of the formula with Gödel number n. We have seen that Q is recursively axiomatized; but so is PA and any reasonable theory whose axioms and rules are recursively described.

T12.17. If T is a recursively axiomatized formal theory then the set of theorems of T is recursively enumerable.

Consider pairs $\langle \mathsf{p}, \mathsf{t}\rangle$ where p numbers a proof of the theorem numbered t, each such pair itself associated with a number, $2^{\mathsf{p}} \times 3^{\mathsf{t}}$. Then there is a recursive function from the integers to these *codes* as follows.

$$\mathsf{code}(0) = \mu z(\exists p < z)(\exists t < z)[z = 2^{p} \times 3^{t} \wedge \text{PRFT}(p, t)]$$
$$\mathsf{code}(Sn) = \mu z(\exists p < z)(\exists t < z)[z > \mathsf{code}(n) \wedge z = 2^{p} \times 3^{t} \wedge \text{PRFT}(p, t)]$$

So 0 is associated with the least integer that codes a proof of a sentence, 1 with the next, and so forth. Then,

$$\mathsf{enum}(n) = \mathsf{exp}(\mathsf{code}(n), 1)$$

returns the Gödel number of theorem n in this ordering.

Recall that π_1 is 3; so $\mathsf{exp}(\mathsf{code}(n), 1)$ returns the number of the proved formula. A given theorem might appear more than once in the enumeration, corresponding to

codes with different proofs of it, but this is no problem, as each theorem appears in some position(s) of the list. Observe that we have, for the first time, made use of regular minimization — so that this function is recursive but not *primitive* recursive. Supposing that T has an infinite number of theorems, there is always some z at which the characteristic function upon which the minimization operates returns zero — so that the function is well-defined. So the theorems of a recursively axiomatized formal theory T are recursively enumerable.

Suppose we add that T is consistent and negation complete. Then there is a recursive relation $\textsc{thrmt}(\mathsf{p})$ true just of numbers for theorems of T: Intuitively, we can enumerate the theorems; then if T is consistent and negation complete, for any sentence $\mathscr{P}$, exactly one of $\mathscr{P}$ or $\sim\mathscr{P}$ must show up in the enumeration. So we can search through the list until we find either $\mathscr{P}$ or $\sim\mathscr{P}$ — and if the one we find is $\mathscr{P}$, then $\mathscr{P}$ is a theorem. In particular, we find $\mathscr{P}$ or $\sim\mathscr{P}$ at the position, $\mu\mathsf{n}[\mathsf{enum}(\mathsf{n}) = \ulcorner\mathscr{P}\urcorner \vee \mathsf{enum}(\mathsf{n}) = \ulcorner\sim\mathscr{P}\urcorner]$. For this, first take,

$$\mathsf{neg}(\mathsf{p}) =_{\text{def}} \ulcorner\sim\urcorner \star \mathsf{p}$$

So if p is the number of a formula $\mathscr{P}$, $\mathsf{neg}(\mathsf{p})$ is the number of $\sim\mathscr{P}$. Now,

T12.18. For any recursively axiomatized, consistent, negation complete formal theory T there is a recursive relation $\textsc{thrmt}(\mathsf{p})$ true just in case p numbers a theorem of T. Set,

$$\mathsf{pos}(\mathsf{p}) = \mu\mathsf{n}([\sim\textsc{sent}(\mathsf{p}) \wedge \mathsf{n} = 0] \vee [\textsc{sent}(\mathsf{p}) \wedge (\mathsf{enum}(\mathsf{n}) = \mathsf{p} \vee \mathsf{enum}(\mathsf{n}) = \mathsf{neg}(\mathsf{p}))])$$

$$\textsc{thrmt}(\mathsf{p}) =_{\text{def}} \mathsf{enum}(\mathsf{pos}(\mathsf{p})) = \mathsf{p}$$

First, $\mathsf{pos}(\mathsf{p})$ takes one of three values: if p does not number a sentence it is just 0; if p appears in the enumeration of theorems it is the position of p; and if $\mathsf{neg}(\mathsf{p})$ appears in the enumeration of theorems, it is the position of $\mathsf{neg}(\mathsf{p})$. Then $\textsc{thrmt}(\mathsf{p})$ is true just in case pos takes the second option — just in case p numbers a sentence and p rather than $\mathsf{neg}(\mathsf{p})$ appears in the enumeration of theorems. Observe that $\mathsf{pos}(\mathsf{p})$ returns 0 both when p does not number a sentence, and when p is the number of the first theorem in the enumeration. But when $\mathsf{pos}(\mathsf{p}) = 0$, $\mathsf{enum}(\mathsf{pos}(\mathsf{p}))$ always numbers the first theorem of the enumeration — so that if p is not the number of a sentence $\textsc{thrmt}(\mathsf{p})$ is false, and when p is the number of the first theorem it is true (as it should be). Again, we appeal to regular minimization. It is only because T is negation complete that the function to which the minimization operator applies is regular. So long as p numbers a sentence, the characteristic function for the second

square brackets is sure to go to zero for one disjunct or the other, and when p does not number a sentence, the function for the first square brackets goes to zero. So the function is well-defined.

Now consider a formula $\mathscr{P}(x)$ with free variable x. The *diagonalization* of $\mathscr{P}$ is the formula $\exists x(x = \overline{\ulcorner \mathscr{P} \urcorner} \wedge \mathscr{P}(x))$. So the diagonalization of $\mathscr{P}$ is true just when $\mathscr{P}$ applies to its own Gödel number. To understand this nomenclature, consider a grid with formulas listed down the left in order of their Gödel numbers and the integer Gödel numbers across the top.

	a	b	c	$\ldots$
$\mathscr{P}_{\mathsf{a}}(x)$	$\boldsymbol{\mathscr{P}_{\mathsf{a}}(\overline{\mathsf{a}})}$	$\mathscr{P}_{\mathsf{a}}(\overline{\mathsf{b}})$	$\mathscr{P}_{\mathsf{a}}(\overline{\mathsf{c}})$	
$\mathscr{P}_{\mathsf{b}}(x)$	$\mathscr{P}_{\mathsf{b}}(\overline{\mathsf{a}})$	$\boldsymbol{\mathscr{P}_{\mathsf{b}}(\overline{\mathsf{b}})}$	$\mathscr{P}_{\mathsf{b}}(\overline{\mathsf{c}})$	
$\mathscr{P}_{\mathsf{c}}(x)$	$\mathscr{P}_{\mathsf{c}}(\overline{\mathsf{a}})$	$\mathscr{P}_{\mathsf{c}}(\overline{\mathsf{b}})$	$\boldsymbol{\mathscr{P}_{\mathsf{c}}(\overline{\mathsf{c}})}$	
$\vdots$				

So, going down the main diagonal, formulas are of the sort $\mathscr{P}_{\mathsf{n}}(\overline{\mathsf{n}})$ where the formula numbered n is applied to its Gödel number n.

Let $\mathsf{num(n)}$ be the Gödel number of the standard numeral for n. So,

$$\mathsf{num(0)} = \ulcorner \emptyset \urcorner$$

$$\mathsf{num(Sy)} = \ulcorner S \urcorner \star \mathsf{num(y)}$$

So num is (primitive) recursive. Now $\mathsf{diag(n)}$ is the Gödel number of the diagonalization of the formula with Gödel number n.

$$\mathsf{diag(n)} =_{\text{def}} \ulcorner \exists x(x = \urcorner \star \mathsf{num(n)} \star \ulcorner \wedge \urcorner \star \mathsf{n} \star \ulcorner) \urcorner$$

Since $\mathsf{diag(n)}$ is recursive, for any theory T extending Q there is a formula $Diag(x, y)$ that captures it. So if $\mathsf{diag(m)} = \mathsf{n}$, then $T \vdash Diag(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ and $T \vdash \forall z[Diag(\overline{\mathsf{m}}, z) \rightarrow z = \overline{\mathsf{n}}]$.

T12.19. Let T be any theory that extends Q. Then for any formula $\mathscr{F}(y)$ containing just the variable y free, there is a sentence $\mathscr{H}$ such that $T \vdash \mathscr{H} \leftrightarrow \mathscr{F}(\overline{\ulcorner \mathscr{H} \urcorner})$. The *Diagonal Lemma*.

Suppose T extends Q; since $\mathsf{diag(n)}$ is recursive, there is a formula $Diag(x, y)$ that captures diag. Let $\mathscr{A}(x) =_{\text{def}} \exists y[\mathscr{F}(y) \wedge Diag(x, y)]$ and $\mathsf{a} = \ulcorner \mathscr{A} \urcorner$, the Gödel number of $\mathscr{A}$. Then set $\mathscr{H} =_{\text{def}} \exists x(x = \overline{\mathsf{a}} \wedge \exists y[\mathscr{F}(y) \wedge Diag(x, y)])$ and $\mathsf{h} = \ulcorner \mathscr{H} \urcorner$, the Gödel number of $\mathscr{H}$. $\mathscr{H}$ is the diagonalization of $\mathscr{A}$; so

$\mathsf{diag}(\mathsf{a}) = \mathsf{h}$. Intuitively, $\mathcal{A}$ says $\mathcal{F}$ applies to the diagonalization of x; so that $\mathcal{H}$ says that $\mathcal{F}$ applies to the diagonalization of $\mathcal{A}$, which is just to say that according to $\mathcal{H}$, $\mathcal{F}(\overline{\ulcorner \mathcal{H} \urcorner})$. Reason as follows.

1.	$\mathcal{H} \leftrightarrow \exists x(x = \overline{\mathsf{a}} \wedge \exists y[\mathcal{F}(y) \wedge Diag(x, y)])$	from def $\mathcal{H}$
2.	$Diag(\overline{\mathsf{a}}, \overline{\mathsf{h}})$	from capture
3.	$\forall z(Diag(\overline{\mathsf{a}}, z) \rightarrow z = \overline{\mathsf{h}})$	from capture
4.	$\quad \mathcal{H}$	A ($g \leftrightarrow$I)
5.	$\quad \exists x(x = \overline{\mathsf{a}} \wedge \exists y[\mathcal{F}(y) \wedge Diag(x, y)])$	1,4 $\leftrightarrow$E
6.	$\quad\quad j = \overline{\mathsf{a}} \wedge \exists y[\mathcal{F}(y) \wedge Diag(j, y)]$	A (g 5$\exists$E)
7.	$\quad\quad j = \overline{\mathsf{a}}$	6 $\wedge$E
8.	$\quad\quad \exists y[\mathcal{F}(y) \wedge Diag(j, y)]$	6 $\wedge$E
9.	$\quad\quad\quad \mathcal{F}(k) \wedge Diag(j, k)$	A (g 8$\exists$E)
10.	$\quad\quad\quad \mathcal{F}(k)$	9 $\wedge$E
11.	$\quad\quad\quad Diag(j, k)$	9 $\wedge$E
12.	$\quad\quad\quad Diag(\overline{\mathsf{a}}, k)$	11,7 =E
13.	$\quad\quad\quad Diag(\overline{\mathsf{a}}, k) \rightarrow k = \overline{\mathsf{h}}$	3 $\forall$E
14.	$\quad\quad\quad k = \overline{\mathsf{h}}$	13,12 $\rightarrow$E
15.	$\quad\quad\quad \mathcal{F}(\overline{\mathsf{h}})$	10,14 =E
16.	$\quad\quad \mathcal{F}(\overline{\mathsf{h}})$	8,9-15 $\exists$E
17.	$\quad \mathcal{F}(\overline{\mathsf{h}})$	5,6-16 $\exists$E
18.	$\quad \mathcal{F}(\overline{\mathsf{h}})$	A $g \leftrightarrow$I
19.	$\quad \mathcal{F}(\overline{\mathsf{h}}) \wedge Diag(\overline{\mathsf{a}}, \overline{\mathsf{h}})$	18,2 $\wedge$I
20.	$\quad \exists y[\mathcal{F}(y) \wedge Diag(\overline{\mathsf{a}}, y)]$	19 $\exists$I
21.	$\quad \overline{\mathsf{a}} = \overline{\mathsf{a}}$	=I
22.	$\quad \overline{\mathsf{a}} = \overline{\mathsf{a}} \wedge \exists y[\mathcal{F}(y) \wedge Diag(\overline{\mathsf{a}}, y)]$	21,20 $\wedge$I
23.	$\quad \exists x(x = \overline{\mathsf{a}} \wedge \exists y[\mathcal{F}(y) \wedge Diag(x, y)])$	22 $\exists$I
24.	$\quad \mathcal{H}$	1,23 $\leftrightarrow$E
25.	$\mathcal{H} \leftrightarrow \mathcal{F}(\overline{\mathsf{h}})$	4-17,18-24 $\leftrightarrow$I
26.	$\mathcal{H} \leftrightarrow \mathcal{F}(\overline{\ulcorner \mathcal{H} \urcorner})$	25 abv

So $T \vdash \mathcal{H} \leftrightarrow \mathcal{F}(\overline{\ulcorner \mathcal{H} \urcorner})$.

If n is such that $\mathsf{f}(\mathsf{n}) = \mathsf{n}$, then n is said to be a *fixed point* for f. And by a (possibly strained) analogy, $\mathcal{H}$ is said to be a “fixed point” for $\mathcal{F}(y)$.

Given things to come, and especially Gödel’s own sentence $\mathcal{G}$ which is *true* though unprovable, it is worth observing that if T is an unsound theory extending Q, then there are false fixed points for $\mathcal{F}$. To see this, recall that if *Diag* captures diag, then so does $Diag \wedge \mathcal{X}$ for any theorem $\mathcal{X}$ — where this remains even if $\mathcal{X}$ is among the theorems that are not true. So, for an unsound theory, let $Diag^*$ be

Diag $\wedge$ $\mathcal{X}$ for any false theorem $\mathcal{X}$, and everything else be the same. Then with *Diag** in place of *Diag*, $T \vdash \mathcal{H}^* \leftrightarrow \mathcal{F}(\overline{\ulcorner \mathcal{H}^* \urcorner})$; but $\mathcal{H}^*$ is not true, insofar as it includes the false conjunct $\mathcal{X}$.

Now we are very close to the incompleteness of arithmetic. As a final preliminary,

T12.20. For no consistent theory T that extends Q is there a recursive relation $\textsc{thrmt}(\mathsf{n})$ that is true just in case n is a Gödel number of a theorem of T.

Consider a consistent theory extending Q; and suppose there is a recursive relation $\textsc{thrmt}(\mathsf{n})$ true just in case n numbers a theorem of T. Since T extends Q and $\textsc{thrmt}$ is recursive, with T12.16 there is some formula *Thrmt*(y) that captures $\textsc{thrmt}$. And again since T extends Q, by the diagonal lemma T12.19, there is a formula $\mathcal{H}$ with Gödel number $\ulcorner \mathcal{H} \urcorner = \mathsf{h}$ such that,[9]

$$T \vdash \mathcal{H} \leftrightarrow \sim Thrmt(\overline{\ulcorner \mathcal{H} \urcorner})$$

Suppose $T \nvdash \mathcal{H}$; then $\mathcal{H}$ is not a theorem of T so that $\mathsf{h} \notin \textsc{thrmt}$; so by capture, $T \vdash \sim Thrmt(\overline{\ulcorner \mathcal{H} \urcorner})$; so by $\leftrightarrow$E, $T \vdash \mathcal{H}$. This is impossible; reject the assumption: $T \vdash \mathcal{H}$. But then $\mathcal{H}$ is a theorem of T; so $\mathsf{h} \in \textsc{thrmt}$; so by capture, $T \vdash Thrmt(\overline{\ulcorner \mathcal{H} \urcorner})$; so by NB, $T \vdash \sim\mathcal{H}$, and T is inconsistent; but by hypothesis, T is consistent. Reject the original assumption: there is no recursive relation $\textsc{thrmt}$.

So from T12.18 any recursively axiomatized, consistent, *negation complete* formal theory has a recursive relation $\textsc{thrmt}(\mathsf{n})$ true just in case n numbers a theorem. But from T12.20 for no consistent theory extending Q is there such a relation. This already suggests results to follow.

*E12.31. Let T be any theory that extends Q. For any formulas $\mathcal{F}_1(y)$ and $\mathcal{F}_2(y)$, generalize the diagonal lemma to find sentences $\mathcal{H}_1$ and $\mathcal{H}_2$ such that,

$$T \vdash \mathcal{H}_1 \leftrightarrow \mathcal{F}_1(\overline{\ulcorner \mathcal{H}_2 \urcorner})$$
$$T \vdash \mathcal{H}_2 \leftrightarrow \mathcal{F}_2(\overline{\ulcorner \mathcal{H}_1 \urcorner})$$

Demonstrate your result. Hint: You will want to generalize the notion of diagonalization so that the *alternation* of formulas $\mathcal{F}_1(z)$, $\mathcal{F}_2(z)$, and $\mathcal{P}$ is

[9] Often $\mathcal{G}$ for Gödel, but this existential variable is not the same as Gödel's constructed sentence; so $\mathcal{H}$, "after" Gödel.

$\exists w \exists x \exists y(w = \overline{\ulcorner \mathcal{P} \urcorner} \wedge x = \overline{\ulcorner \mathcal{F}_2 \urcorner} \wedge y = \overline{\ulcorner \mathcal{F}_1 \urcorner} \wedge \exists z(\mathcal{F}_1(z) \wedge \mathcal{P}))$. Then you can find a recursive function $\mathsf{alt}(\mathsf{p}, \mathsf{f}_1, \mathsf{f}_2)$ whose output is the number of the alternation of formulas numbered p, f_1 and f_2, where this function is captured by some formula $Alt(w, x, y, z)$ that itself has Gödel number a. Then $\mathsf{alt}(\overline{\mathsf{a}}, \overline{\mathsf{f}}_1, \overline{\mathsf{f}}_2)$ and $\mathsf{alt}(\overline{\mathsf{a}}, \overline{\mathsf{f}}_2, \overline{\mathsf{f}}_1)$ number the formulas you need for $\mathcal{H}_1$ and $\mathcal{H}_2$.

E12.32. Use your version of the diagonal lemma from E12.31 to provide an alternate demonstration of T12.20. Hint: You will be able to set up sentences such that the first says the second is not a theorem, while the second says the first is a theorem.

12.5.2 First Applications

Here are three quick results from our theorems. Do not let the simplicity of their proof (if the proof can seem simple after all we have done) distract from the significance of their content!

The Incompleteness of Arithmetic.

T12.21. No consistent, recursively axiomatizable theory extending Q is negation complete. The *incompleteness of arithmetic*.

Consider a theory T that is a consistent, recursively axiomatizable extension of Q. Then since T consistent and extends Q, by T12.20, there is no recursive relation $\mathsf{THRMT}(\mathsf{n})$ true iff n is the Gödel number of a theorem. Suppose T is negation complete; then since T is also consistent and recursively axiomatized, by T12.18 there is a recursive relation $\mathsf{THRMT}(\mathsf{n})$ true iff n is the Gödel number of a theorem. This is impossible, reject the assumption: T is not negation complete.

It immediately follows that Q and PA are not negation complete. But similarly for *any* consistent recursively axiomatizable theory that extends Q. We already knew that there were formulas $\mathcal{P}$ such that Q $\nvdash \mathcal{P}$ and Q $\nvdash \sim\mathcal{P}$. But we did not already have this result for PA; and we certainly did not have the result generally for recursively axiomatizable theories extending Q.

There are other ways to obtain this result. We explore Gödel's own strategy in the next chapter. And we shall see an approach from computability in chapter 14. However, this first argument is sufficient to establish the point.

The Decision Problem

It is a short step from the result that if Q is consistent, then no recursive relation identifies the theorems of Q, to the result that if Q is consistent, then no recursive relation identifies the theorems of predicate logic.

T12.22. If Q is consistent, then no recursive relation $\textsc{thrmpl}(\mathsf{n})$ is true iff n numbers a theorem of predicate logic.

Suppose otherwise, that Q is consistent and some recursive relation $\textsc{thrmpl}(\mathsf{n})$ is true iff n numbers a theorem of predicate logic. Let $\mathcal{Q}$ be the conjunction of the axioms of Q; then $\mathcal{P}$ is a theorem of Q iff $\vdash \mathcal{Q} \to \mathcal{P}$. Let $\mathsf{q} = \ulcorner\mathcal{Q}\urcorner$; then,

$$\textsc{thrmq}(\mathsf{n}) =_{\text{def}} \textsc{thrmpl}(\mathsf{q} \star \ulcorner\to\urcorner \star \mathsf{n})$$

defines a recursive function true iff n numbers a theorem of Q. But, given the consistency of Q, by T12.20, there is no function $\textsc{thrmq}(\mathsf{n})$. Reject the assumption, if Q is consistent, then there is no recursive relation $\textsc{thrmpl}(\mathsf{n})$ true iff n numbers a theorem of predicate logic.

And, of course, given that Q *is* consistent, it follows that no recursive relation numbers the theorems of predicate logic. From T12.20 no recursive relation numbers the theorems of Q. Now we see that this result extends to the theorems of predicate logic. At at this stage, these results may seem to be a sort of curiosity about what recursive functions do. They gain significance when, as we have already hinted can be done, we identify the recursive functions with the *computable* functions in chapter 14.

Tarski's Theorems

A couple of related theorems fall under this heading. Say $\textsc{true}(\mathsf{n})$ is true iff n numbers a true sentence of some language $\mathcal{L}$. We do not assume that $\textsc{true}(\mathsf{n})$ is recursive — only that, by definition, it applies to numbers of true sentences. Suppose $True(x)$ expresses $\textsc{true}(\mathsf{n})$. Then by expression, $\mathsf{I}[True(\overline{\ulcorner\mathcal{P}\urcorner})] = \mathsf{T}$ iff $\ulcorner\mathcal{P}\urcorner \in \textsc{true}$; and this iff $\mathsf{I}[\mathcal{P}] = \mathsf{T}$. So, with some manipulation,

$$\mathsf{I}[True(\overline{\ulcorner\mathcal{P}\urcorner}) \leftrightarrow \mathcal{P}] = \mathsf{T}$$

Let us say T is a *truth theory* for language $\mathcal{L}$, iff for any sentence of $\mathcal{L}$, T proves this result.

$$T \vdash True(\overline{\ulcorner \mathcal{P} \urcorner}) \leftrightarrow \mathcal{P}$$

Nothing prevents theories of this sort. However, a first theorem is to the effect that theories in our range cannot be theories of truth for their own language $\mathcal{L}$.

T12.23. No recursively axiomatized consistent theory extending Q is a theory of truth for its own language $\mathcal{L}$.

Suppose otherwise, that a recursively axiomatized consistent T extending Q is a theory of truth for its own $\mathcal{L}$. Since T extends Q, by the diagonal lemma, there is a sentence $\mathcal{F}$ (a false or liar sentence) such that

$$T \vdash \mathcal{F} \leftrightarrow \sim True(\overline{\ulcorner \mathcal{F} \urcorner})$$

But since T is a truth theory, $T \vdash True(\overline{\ulcorner \mathcal{F} \urcorner}) \leftrightarrow \mathcal{F}$; so $T \vdash True(\overline{\ulcorner \mathcal{F} \urcorner}) \leftrightarrow \sim True(\overline{\ulcorner \mathcal{F} \urcorner})$; so T is inconsistent. Reject the assumption: T is not a truth theory for its language $\mathcal{L}$.

This theorem explains our standard jump to the metalanguage when we give conditions like ST and SF. Nothing prevents stating truth conditions — trouble results when a theory purports to give conditions for all the sentences in its own language.

A second theorem takes on the slightly stronger (but still plausible) assumption that Q is a sound theory, so that all of its theorems are true. Under this condition, there is trouble even expressing a truth predicate for language $\mathcal{L}$ in that language $\mathcal{L}$.

T12.24. If Q is sound, and $\mathcal{L}$ includes $\mathcal{L}_{NT}$ then there is no *True* to express TRUE in $\mathcal{L}$.

Suppose otherwise, that Q is sound and some formula $True(x)$ expresses $\textsf{TRUE}(\mathsf{n})$ in $\mathcal{L}$; since Q is a theory that extends Q, by the diagonal lemma, there is a sentence $\mathcal{F}$ such that $\mathrm{Q} \vdash \mathcal{F} \leftrightarrow \sim True(\overline{\ulcorner \mathcal{F} \urcorner})$; since the theorems of Q are true, $\mathsf{N}[\mathcal{F} \leftrightarrow \sim True(\overline{\ulcorner \mathcal{F} \urcorner})] = \mathsf{T}$; so with a bit of manipulation,

$$\mathsf{N}[\mathcal{F}] = \mathsf{T} \text{ iff } \mathsf{N}[\sim True(\overline{\ulcorner \mathcal{F} \urcorner})] = \mathsf{T}; \text{ iff } \mathsf{N}[True(\overline{\ulcorner \mathcal{F} \urcorner})] \neq \mathsf{T}$$

(i) Suppose $\mathsf{N}[True(\overline{\ulcorner \mathcal{F} \urcorner})] \neq \mathsf{T}$; then by expression, $\ulcorner \mathcal{F} \urcorner \notin \textsf{TRUE}$, so that $\mathsf{N}[\mathcal{F}] \neq \mathsf{T}$; so by the above equivalence, $\mathsf{N}[True(\overline{\ulcorner \mathcal{F} \urcorner}) = \mathsf{T}$; reject the assumption. (ii) So $\mathsf{N}[True(\overline{\ulcorner \mathcal{F} \urcorner}) = \mathsf{T}$; but then by the equivalence, $\mathsf{N}[\mathcal{F}] \neq \mathsf{T}$; so $\ulcorner \mathcal{F} \urcorner \notin \textsf{TRUE}$; so by expression, $\mathsf{N}[\sim True(\overline{\ulcorner \mathcal{F} \urcorner})] = \mathsf{T}$; so $\mathsf{N}[True(\overline{\ulcorner \mathcal{F} \urcorner}) \neq \mathsf{T}$; this is impossible.

Reject the original assumption: no formula $True(x)$ expresses $\textsf{TRUE}(\mathsf{n})$.

Observe that some numerical properties are both expressed and captured — as the recursive relations. As we have seen, though $\textsc{thrmq}(\mathsf{n})$ is a relation on the integers, it is not not a recursive relation. It can however be *expressed* by the formula, $\exists x Prfq(x, n)$. Then, once we show (in T14.10) that all the functions captured by a recursively axiomatized consistent theory extending Q are recursive, it follows that $\textsc{thrmq}(\mathsf{n})$ is expressed but not captured. And now we have seen a relation $\textsc{true}(\mathsf{n})$ not even expressed in $\mathscr{L}_{NT}$.

This is a decent start into the results of Part IV of the text. In the following, we turn to deepening and extending them in different directions.

Final Results of Chapter 12

T12.17 If T is a recursively axiomatized formal theory then the set of theorems of T is recursively enumerable.

T12.18 For any recursively axiomatized, consistent, negation complete formal theory T there is a recursive relation $\textsc{thrmt}(\mathsf{n})$ true just in case n numbers a theorem of T.

T12.19 Let T be any theory that extends Q. Then for any formula $\mathcal{F}(y)$ containing just the variable y free, there is a sentence $\mathcal{H}$ such that $T \vdash \mathcal{H} \leftrightarrow \mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})$. The *Diagonal Lemma*.

T12.20 For no consistent theory T that extends Q is there a recursive relation $\textsc{thrmt}(\mathsf{n})$ that is true just in case n is a Gödel number of a theorem of T.

T12.21 No consistent, recursively axiomatizable extension of Q is negation complete. The *incompleteness of arithmetic*.

T12.22 If Q is consistent, then no recursive relation $\textsc{thrmpl}(\mathsf{n})$ is true iff n numbers a theorem of predicate logic

T12.23 No recursively axiomatized consistent theory extending Q is a theory of truth for its own language $\mathscr{L}$.

T12.24 If Q is sound, and $\mathscr{L}$ includes $\mathscr{L}_{NT}$ then there is no *True* to express $\textsc{true}$ in $\mathscr{L}$.

E12.33. Use the alternate version of the diagonal lemma from E12.31 to provide alternate demonstrations of T12.23 and T12.24. Include the “bit of minipulation” left out of the text for T12.24.

E12.34. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The recursive functions and the role of the beta function in their expression and capture.

b. The essential elements from this chapter contributing to the proof of the incompleteness of arithmetic.

c. The essential elements from this chapter contributing to the proof of that no recursive relation identifies the theorems of predicate logic

d. The essential elements from this chapter contributing to the proof of Tarski's theorem.

Chapter 13

Gödel's Theorems

We have seen a demonstration of the incompleteness of arithmetic. In this chapter, we take another run at that result, this time by Gödel's original strategy of producing sentences that are true iff not provable. This enables us to extend and deepen the incompleteness result, and puts us in a position to take up Gödel's second incompleteness theorem, according to which theories (of a certain sort) are not sufficient for demonstrations of *consistency*.

13.1 Gödel's First Theorem

Recall that the diagonalization of a formula $\mathscr{P}(x)$ is $\exists x(x = \overline{\ulcorner \mathscr{P} \urcorner} \wedge \mathscr{P}(x))$. In addition, there is a recursive function $\mathsf{diag}(\mathsf{n})$ which numbers the diagonalization of the formula with number n and, if T is recursively axiomatized, a recursive relation $\textsc{prft}(\mathsf{m}, \mathsf{n})$ true when m numbers a proof of the formula with number n. Our previous argument for incompleteness required $\textsc{prft}(\mathsf{m}, \mathsf{n})$ for T12.17, and a $\mathit{Diag}(x, y)$ to capture $\mathsf{diag}(\mathsf{n})$ for the diagonal lemma. Previously, under the assumption that there is a $\textsc{thrmt}$ and so *Thrmt* we applied the diagonal lemma so that $T \vdash \mathscr{H} \leftrightarrow {\sim}\mathit{Thrmt}(\overline{\ulcorner \mathscr{H} \urcorner})$ to reach contradiction, and argued that there must be a sentence such that neither it nor its negation is provable — without any suggestion what that sentence might be. This time, by related methods, we construct a particular sentence such that neither it nor its negation is provable.

13.1.1 Semantic Version

Consider some recursively axiomatized theory T whose language includes $\mathscr{L}_{\text{NT}}$. Since $\textsc{prft}(\mathsf{m}, \mathsf{n})$ and $\mathsf{diag}(\mathsf{n})$ are recursive, they are *expressed* by some formulas $\mathit{Prft}(x, y)$

and $Diag(x, y)$. Let $\mathscr{A}(z) =_{\text{def}} \sim\exists x\exists y(Prft(x, y) \wedge Diag(z, y))$, and $\mathsf{a} = \ulcorner\mathscr{A}\urcorner$. So $\mathscr{A}$ says nothing numbers a proof of the diagonalization of a formula with number z. Then,

$$\mathscr{G} =_{\text{def}} \exists z(z = \overline{\mathsf{a}} \wedge \sim\exists x\exists y(Prft(x, y) \wedge Diag(z, y)))$$

So $\mathscr{G}$ is the diagonalization of $\mathscr{A}$, and intuitively $\mathscr{G}$ "says" that nothing numbers a proof of it. Let $\mathsf{g} = \ulcorner\mathscr{G}\urcorner$. Observe that $\mathscr{G}$ is defined relative to *Prft* for T; so each T yields its own Gödel sentence (if it were not ugly, we might sensibly introduce subscripts $\mathscr{G}_T$). Thus,

T13.1. For any recursively axiomatized theory T whose language includes $\mathscr{L}_{NT}$, $\mathscr{G}$ is true iff it is unprovable in T (iff $T \nvdash \mathscr{G}$).

> Consider a recursively axiomatized theory T whose language includes $\mathscr{L}_{NT}$ and $\mathscr{G}$ as described above. Skipping some steps, (i) Suppose $\mathsf{N}[\mathscr{G}] = \mathsf{T}$; then for any d, $\mathsf{N_d}[\mathscr{G}] = \mathsf{S}$; so with T10.2, $\mathsf{N_d}[\sim\exists x\exists y(Prft(x, y) \wedge Diag(\overline{\mathsf{a}}, y))] = \mathsf{S}$; so that there are no m, n such that $\mathsf{N}[Prft(\overline{\mathsf{m}}, \overline{\mathsf{n}})] = \mathsf{T}$ and $\mathsf{N}[Diag(\overline{\mathsf{a}}, \overline{\mathsf{n}})] = \mathsf{T}$; so by expression, there are no m, n such that $\langle \mathsf{m}, \mathsf{n}\rangle \in \textsc{prft}$ and $\langle \mathsf{a}, \mathsf{n}\rangle \in \mathsf{diag}$; but $\mathsf{diag}(\mathsf{a}) = \mathsf{g}$; so no m numbers a proof of $\mathscr{G}$, which is to say $T \nvdash \mathscr{G}$. (ii) Suppose $\mathsf{N}[\mathscr{G}] \neq \mathsf{T}$; then there is some d such that $\mathsf{N_d}[\mathscr{G}] \neq \mathsf{S}$ and for any $\mathsf{n} \in \mathbb{N}$, $\mathsf{N_{d(z|n)}}[z = \overline{\mathsf{a}} \wedge \sim\exists x\exists y(Prft(x, y) \wedge Diag(z, y))] \neq \mathsf{S}$; in particular, $\mathsf{N_{d(z|a)}}[z = \overline{\mathsf{a}} \wedge \sim\exists x\exists y(Prft(x, y) \wedge Diag(z, y))] \neq \mathsf{S}$; so with T10.2, $\mathsf{N_d}[\sim\exists x\exists y(Prft(x, y) \wedge Diag(\overline{\mathsf{a}}, y))] \neq \mathsf{S}$ and $\mathsf{N_d}[\exists x\exists y(Prft(x, y) \wedge Diag(\overline{\mathsf{a}}, y))] = \mathsf{S}$; so there are m and n such that both $Prft(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ and $Diag(\overline{\mathsf{a}}, \overline{\mathsf{n}})$ are satisfied on N with d; so $\mathsf{N}[\sim Prft(\overline{\mathsf{m}}, \overline{\mathsf{n}})] \neq \mathsf{T}$ and $\mathsf{N}[\sim Diag(\overline{\mathsf{a}}, \overline{\mathsf{n}})] \neq \mathsf{T}$; and by expression $\langle \mathsf{m}, \mathsf{n}\rangle \in \textsc{prft}$ and $\langle \mathsf{a}, \mathsf{n}\rangle \in \mathsf{diag}$; but again, $\mathsf{diag}(\mathsf{a}) = \mathsf{g}$; so $\langle \mathsf{m}, \mathsf{g}\rangle \in \textsc{prft}$; so $T \vdash \mathscr{G}$; so by transposition, if $T \nvdash \mathscr{G}$, then $\mathsf{N}[\mathscr{G}] = \mathsf{T}$.

It is not a difficult exercise to fill in the details. Intuitively this result should seem right. Suppose $\mathscr{G}$ "says" that it is unprovable: then if it is true it is unprovable; and if it is unprovable it is true; so it is true iff it is unprovable.

Now suppose that T is recursively axiomatized, and *sound* theory (so that its theorems are true), whose language includes $\mathscr{L}_{NT}$. Then T is negation incomplete.

T13.2. If T is a recursively axiomatized sound theory whose language includes $\mathscr{L}_{NT}$, then T is negation incomplete.

> Suppose T is a recursively axiomatized theory whose language includes $\mathscr{L}_{NT}$; then there is a sentence $\mathscr{G}$ to which the conditions for T13.1 apply. (i) Suppose

$T \vdash \mathcal{G}$; then, since T is sound, $\mathcal{G}$ is true; so by T13.1, $T \nvdash \mathcal{G}$; reject the assumption, $T \nvdash \mathcal{G}$. Suppose $T \vdash \sim\mathcal{G}$; then since T is sound, $\sim\mathcal{G}$ is true; so $\mathcal{G}$ is not true; so by T13.1, $T \vdash \mathcal{G}$; so by soundness again, $\mathcal{G}$ is true; reject the assumption: $T \nvdash \sim\mathcal{G}$.

So $\mathcal{G}$ is a sentence such that if T is a recursively axiomatized sound theory whose language includes $\mathcal{L}_{\text{NT}}$, neither $\mathcal{G}$ nor its negation is a theorem. And, from T13.1, given that $\mathcal{G}$ is unprovable, if T is a recursively axiomatized theory whose language includes $\mathcal{L}_{\text{NT}}$, then $\mathcal{G}$ is a *true* non-theorem. This version of the incompleteness result depends on the ability to express $\mathcal{G}$, together with the soundness of theory T.

13.1.2 Syntactic Version

Gödel's first theorem is usually presented with the capture and consistency, rather than the expression and soundness constraints. We turn now to a version of this first sort which, again, builds a particular sentence such that neither it nor its negation is provable.

Since $\textsf{PRFT}(\mathsf{m},\mathsf{n})$ and $\mathsf{diag}(\mathsf{n})$ are recursive, in theories extending Q they are *captured* by canonical formulas $Prft(x,y)$ and $Diag(x,y)$. As before, let $\mathcal{A}(z) =_{\text{def}} \sim\exists x\exists y(Prft(x,y) \wedge Diag(z,y))$, and $\mathsf{a} = \ulcorner\mathcal{A}\urcorner$. So $\mathcal{A}$ says nothing numbers a proof of the diagonalization of a formula with number z. Then,

$$\mathcal{G} =_{\text{def}} \exists z(z = \overline{\mathsf{a}} \wedge \sim\exists x\exists y(Prft(x,y) \wedge Diag(z,y)))$$

So $\mathcal{G}$ is the diagonalization of $\mathcal{A}$; and let g be the Gödel number of $\mathcal{G}$. This time, we shall be able to prove the relation between $\mathcal{G}$ and a proof of it. Reasoning as for the diagonal lemma,

T13.3. Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathcal{G} \leftrightarrow \sim\exists x Prft(x, \overline{\ulcorner\mathcal{G}\urcorner})$.

Since T is recursively axiomatized, there is a recursive $\textsf{PRFT}$ and since T extends Q there are *Prft* and *Diag* that capture $\textsf{PRFT}$ and diag. From the definition of $\mathcal{G}$, $T \vdash \mathcal{G} \leftrightarrow \exists z(z = \overline{\mathsf{a}} \wedge \sim\exists x\exists y[Prft(x,y) \wedge Diag(z,y)])$; from capture $T \vdash Diag(\overline{\mathsf{a}},\overline{\mathsf{g}})$; and $T \vdash \forall z(Diag(\overline{\mathsf{a}},z) \rightarrow z = \overline{\mathsf{g}})$. From these it follows that $T \vdash \mathcal{G} \leftrightarrow \sim\exists x Prft(x,\overline{\mathsf{g}})$; which is to say, $T \vdash \mathcal{G} \leftrightarrow \sim\exists x Prft(x, \overline{\ulcorner\mathcal{G}\urcorner})$ (homework).

From the diagonal lemma, under appropriate conditions, given a formula $\mathcal{F}(y)$, there is some $\mathcal{H}$ such that $T \vdash \mathcal{H} \leftrightarrow \mathcal{F}(\overline{\ulcorner\mathcal{H}\urcorner})$. Under the assumption that there is $\textsf{THRMT}$,

we applied this to show there would be some $\mathscr{H}$ such that $T \vdash \mathscr{H} \leftrightarrow \sim Thrmt(\overline{\ulcorner\mathscr{H}\urcorner})$. This led to contradiction. In this case, however, we show that there really is a particular sentence $\mathscr{G}$ such that $T \vdash \mathscr{G} \leftrightarrow \sim\exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$.

Our idea is to show that if T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathscr{G}$ and $T \nvdash \sim\mathscr{G}$. The first is easy enough.

T13.4. If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathscr{G}$.

> Suppose T is a consistent recursively axiomatized theory extending Q. Suppose $T \vdash \mathscr{G}$; then since T is recursively axiomatized, for some m, $\textsc{prft}(\mathsf{m}, \mathsf{g})$; and since T extends Q, by capture, $T \vdash Prft(\overline{\mathsf{m}}, \overline{\mathsf{g}})$; so by $\exists$I, $T \vdash \exists x Prft(x, \overline{\mathsf{g}})$, which is to say, $T \vdash \exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$. But since $T \vdash \mathscr{G}$, by T13.3, $T \vdash \sim\exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$. So T is inconsistent; reject the assumption: $T \nvdash \mathscr{G}$.

That is the first half of what we are after. But we can't quite get that if T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \sim\mathscr{G}$. Rather, we need a strengthened notion of consistency. Say a theory T is *ω-incomplete* iff for some $\mathscr{P}(x)$, T can prove each $\mathscr{P}(\overline{\mathsf{m}})$ but T cannot go on to prove $\forall x \mathscr{P}(x)$. Equivalently, T is ω-incomplete iff for every m, it can prove each $T \vdash \sim\mathscr{P}(\overline{\mathsf{m}})$ but $T \nvdash \sim\exists x \mathscr{P}(x)$. We have seen that Q is ω-incomplete: we can prove, say $\overline{\mathsf{n}} \times \overline{\mathsf{m}} = \overline{\mathsf{m}} \times \overline{\mathsf{n}}$ for every m and n, but cannot go on to prove the corresponding universal generalization, $\forall x \forall y (x \times y = y \times x)$. Say T is *ω-inconsistent* iff for some $\mathscr{P}(x)$, T proves each $\mathscr{P}(\overline{\mathsf{m}})$ but also proves $\sim\forall x \mathscr{P}(x)$. Equivalently, T is ω-inconsistent iff for every m, can prove each $T \vdash \sim\mathscr{P}(\overline{\mathsf{m}})$ and $T \vdash \exists x \mathscr{P}(x)$. ω-incompleteness is a theoretical weakness — there are some things true but not provable. But ω-inconsistency is a theoretical disaster: It is not possible for the theorems of an ω-inconsistent theory all to be true on any interpretation (assuming some $\overline{\mathsf{m}}$ for each $\mathsf{m} \in \mathsf{U}$). ω-inconsistency is not itself inconsistency — for we do not have any sentence such that $T \vdash \mathscr{P}$ and $T \vdash \sim\mathscr{P}$. But inconsistent theories are automatically ω-inconsistent — for from contradiction all consequences follow (including each $\mathscr{P}(\overline{\mathsf{m}})$ and also $\sim\forall x \mathscr{P}(x)$) so that an ω-consistent theory is consistent. Now we show,

T13.5. If T is an ω-consistent, recursively axiomatized theory extending Q, then $T \nvdash \sim\mathscr{G}$.

> Suppose T is an ω-consistent recursively axiomatized theory extending Q. Suppose $T \vdash \sim\mathscr{G}$; if T is ω-consistent, then it is consistent, so $T \nvdash \mathscr{G}$; so since T is recursively axiomatized, for all m, $\langle \mathsf{m}, \mathsf{g} \rangle \notin \textsc{prft}$; and since T extends Q, by capture, $T \vdash \sim Prft(\overline{\mathsf{m}}, \overline{\mathsf{g}})$; and since T is ω-consistent, $T \nvdash \exists x Prft(x, \overline{\mathsf{g}})$; which is to say, $T \nvdash \exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$. But since $T \vdash$

$\sim\mathscr{G}$, by T13.3 with NB, $T \vdash \exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$. This is impossible; reject the assumption: $T \nvdash \sim\mathscr{G}$.

So if a recursively axiomatized theory extending Q has the relevant *consistency* properties, then it is negation incomplete. Further, insofar as T canonically captures the recursive functions, it expresses the recursive functions; so by T13.1, $\mathscr{G}$ is true iff $T \nvdash \mathscr{G}$. So if T is a consistent recursively axiomatized theory extending Q, then $\mathscr{G}$ is both unprovable and true.[1]

This is roughly the form in which Gödel proved the incompleteness of arithmetic in 1931: If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathscr{G}$; and if T is an ω-consistent, recursively axiomatized theory extending Q, then $T \nvdash \sim\mathscr{G}$. Since we believe that standard theories including Q and PA are consistent and ω-consistent, this sufficient for the incompleteness of arithmetic.

E13.1. Fill in the details for the argument of T13.1.

*E13.2. Complete the demonstration of T13.3 by providing a derivation to show $T \vdash \mathscr{G} \leftrightarrow \sim\exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$. The demonstration for the diagonal lemma theorem is a model, though steps will be adapted to the particular form of these sentences.

13.1.3 Rosser's Sentence

But it is possible to drop the special assumption of ω-consistency by means of a sentence somewhat different from $\mathscr{G}$.[2] Recall that $\mathsf{neg}(\mathsf{n})$ is the Gödel number of the negation of the sentence with number n. So $\overline{\mathsf{PRFT}}(\mathsf{m}, \mathsf{n}) =_{\text{def}} \mathsf{PRFT}(\mathsf{m}, \mathsf{neg}(\mathsf{n}))$ obtains when m numbers a proof of the negation of the sentence numbered n. Since it is recursive, it is captured by some $\overline{Prft}(x, y)$. Set,

$$RPrft(x, y) =_{\text{def}} Prft(x, y) \wedge (\forall w \leq x)\sim\overline{Prft}(w, y)$$

So $RPrft(x, y)$ just in case x numbers a proof of the sentence numbered y and no number less than or equal to x is a proof of the negation of that sentence. Now, working as before, set $\mathscr{A}'(z) =_{\text{def}} \sim\exists x\exists y(RPrft(x, y) \wedge Diag(z, y))$, and $\mathsf{a} = \ulcorner\mathscr{A}'\urcorner$. So $\mathscr{A}'$ says nothing numbers an R-proof of the diagonalization of a formula with number z. Then,

[1]Given that an unsound theory has false fixed points, here is another reason to distinguish this constructed $\mathscr{G}$ from the variable $\mathscr{H}$ of the previous chapter. See p. 603n9.

[2]Barkley Rosser, "Extensions of Some Theorems of Gödel and Church."

$$\mathcal{R} =_{\text{def}} \exists z(z = \overline{\mathsf{a}} \land \sim\exists x\exists y(RPrft(x, y) \land Diag(z, y)))$$

So $\mathcal{R}$ is the diagonalization of $\mathcal{A}'$; let r be the Gödel number of $\mathcal{R}$. And $\mathcal{R}$ has the key syntactic property just like $\mathcal{G}$. Again, reasoning as we did for the diagonal lemma,

T13.6. Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathcal{R} \leftrightarrow \sim\exists xRPrft(x, \overline{\ulcorner\mathcal{R}\urcorner})$.

You can show this just as for T13.3.

Now the first half of the incompleteness result is straightforward.

T13.7. If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathcal{R}$.

Suppose T is a consistent recursively axiomatized theory extending Q. Suppose $T \vdash \mathcal{R}$; then since T is recursively axiomatized, for some m, $\textsf{PRFT}(\mathsf{m}, \mathsf{r})$; and since T extends Q, by capture, $T \vdash Prft(\overline{\mathsf{m}}, \overline{\mathsf{r}})$. But by consistency, $T \nvdash \sim\mathcal{R}$; so for all n, and in particular all $\mathsf{n} \leq \mathsf{m}$, $\langle \mathsf{n}, \mathsf{r}\rangle \notin \overline{\textsf{PRFT}}$; so by capture, $T \vdash \sim\overline{Prft}(\overline{\mathsf{n}}, \overline{\mathsf{r}})$; so by T8.21, $T \vdash (\forall w \leq \overline{\mathsf{m}})\sim\overline{Prft}(w, \overline{\mathsf{r}})$; so $T \vdash Prft(\overline{\mathsf{m}}, \overline{\mathsf{r}}) \land (\forall w \leq \overline{\mathsf{m}})\sim\overline{Prft}(w, \overline{\mathsf{r}})$; so $T \vdash RPrft(\overline{\mathsf{m}}, \overline{\mathsf{r}})$; so $T \vdash \exists xRPrft(x, \overline{\mathsf{r}})$, which is to say, $T \vdash \exists xRPrft(x, \overline{\ulcorner\mathcal{R}\urcorner})$. But since $T \vdash \mathcal{R}$, by T13.6, $T \vdash \sim\exists xRPrft(x, \overline{\ulcorner\mathcal{R}\urcorner})$; so T is inconsistent. This is impossible; reject the assumption: $T \nvdash \mathcal{R}$.

So, with consistency, it is not much harder to prove $T \vdash \exists xRPrft(x, \overline{\ulcorner\mathcal{R}\urcorner})$ from the assumption that $T \vdash \mathcal{R}$ than to prove $T \vdash \exists xPrft(x, \overline{\ulcorner\mathcal{G}\urcorner})$ from the assumption that $T \vdash \mathcal{G}$.

Reasoning for the other direction is somewhat more involved, but still straightforward.

T13.8. If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \sim\mathcal{R}$.

Suppose T is a consistent recursively axiomatized theory extending Q. Suppose $T \vdash \sim\mathcal{R}$. Then since T is recursively axiomatized, for some m, $\langle \mathsf{m}, \mathsf{r}\rangle \in \overline{\textsf{PRFT}}$; and since T extends Q, by capture, $T \vdash \overline{Prft}(\overline{\mathsf{m}}, \overline{\mathsf{r}})$. By consistency, $T \nvdash \mathcal{R}$; so for any n, and in particular, any $\mathsf{n} \leq \mathsf{m}$, $\langle \mathsf{n}, \mathsf{r}\rangle \notin \textsf{PRFT}$; so by capture, $T \vdash \sim Prft(\overline{\mathsf{n}}, \overline{\mathsf{r}})$; and by T8.21, $T \vdash (\forall w \leq \overline{\mathsf{m}})\sim Prft(w, \overline{\mathsf{r}})$. Now reason as follows.

1.	$\sim\mathscr{R}$	from T
2.	$\overline{Prft}(\overline{\mathsf{m}},\bar{\mathsf{r}})$	capture
3.	$(\forall w \leq \overline{\mathsf{m}})\sim Prft(w,\bar{\mathsf{r}})$	capture and T8.21
4.	$\mathscr{R} \leftrightarrow \sim\exists x RPrft(x,\bar{\mathsf{r}})$	from T13.6
5.	$\exists x RPrft(x,\bar{\mathsf{r}})$	1,4 NB
6.	$\exists x[Prft(x,\bar{\mathsf{r}}) \wedge (\forall w \leq x)\sim\overline{Prft}(w,\bar{\mathsf{r}})]$	5 abv
7.	$Prft(j,\bar{\mathsf{r}}) \wedge (\forall w \leq j)\sim\overline{Prft}(w,\bar{\mathsf{r}})]$	A (g, 6∃E)
8.	$j \leq \overline{\mathsf{m}} \vee \overline{\mathsf{m}} \leq j$	T8.19
9.	$j \leq \overline{\mathsf{m}}$	A (g 8∨E)
10.	$Prft(j,\bar{\mathsf{r}})$	7 ∧ E
11.	$\sim Prft(j,\bar{\mathsf{r}})$	3,9 (∀)E
12.	$\bot$	10,11 ⊥I
13.	$\overline{\mathsf{m}} \leq j$	A (g, 8∨E)
14.	$(\forall w \leq j)\sim\overline{Prft}(w,\bar{\mathsf{r}})]$	7 ∧E
15.	$\sim\overline{Prft}(\overline{\mathsf{m}},\bar{\mathsf{r}})$	14,13 (∀E)
16.	$\bot$	2,15 ⊥I
17.	$\bot$	8,9-12,13-16 ∨E
18.	$\bot$	6,7-17 ∃E

So $T \vdash \bot$, that is $T \vdash Z \wedge \sim Z$ and T is inconsistent. Reject the assumption, $T \nvdash \sim\mathscr{R}$.

In the previous case, with $\mathscr{G}$, we had no way to convert $\exists x Prft(x,\overline{\mathsf{g}})$ to a contradiction with $\sim Prft(\overline{0},\overline{\mathsf{g}})$, $\sim Prft(\overline{1},\overline{\mathsf{g}})\ldots$; that is why we needed ω-consistency. In this case, the special nature of $\mathscr{R}$ aids the argument: From $\exists x RPrft(x,\bar{\mathsf{r}})$, consider a j such that $RPrft(j,\bar{\mathsf{r}})$. If $j \leq \mathsf{m}$, there is contradiction insofar as we are in the scope of the bounded universal quantifier $(\forall w \leq \overline{\mathsf{m}})\sim Prft(w,\bar{\mathsf{r}})$. If $\mathsf{m} \leq j$, then we end up with both $\overline{Prft}(\overline{\mathsf{m}},\bar{\mathsf{r}})$ and $\sim\overline{Prft}(\overline{\mathsf{m}},\bar{\mathsf{r}})$, as $RPrft(j,\bar{\mathsf{r}})$ builds in inconsistency with $\overline{Prft}(\overline{\mathsf{m}},\bar{\mathsf{r}})$. So $T \nvdash \mathscr{R}$ and $T \nvdash \sim\mathscr{R}$

Let us close this section with some reflections on what we have shown. First,

$$\text{Q is sound} \Longrightarrow \text{Q is } \omega\text{-consistent} \Longrightarrow \text{Q is consistent}$$

So our results are progressively stronger, as the assumptions have become correspondingly weaker. Of course,

$$\text{canonical capture} \Longrightarrow \text{canonical expression}$$

So the second requirement is increased as we move from expression to capture.

Second, we have not shown that there are truths of $\mathscr{L}_{NT}$ not provable in any recursively axiomatizable, consistent theory extending Q. Rather, what we have shown

is that for any recursively axiomatizable consistent theory extending Q, there are some truths of $\mathscr{L}_{NT}$ not provable in that theory. For a given recursively axiomatizable theory, there will be a given relation $\textsc{prft}(\mathsf{m}, \mathsf{n})$ and $Prft(x, y)$ depending on the particular axioms of that theory — and so unique sentences $\mathscr{G}$ and $\mathscr{R}$ constructed as above. In particular, given that a theory cannot prove, say, $\mathscr{R}$, we might simply *add* $\mathscr{R}$ to its axioms; then of course there is a derivation of $\mathscr{R}$ from the axioms of the revised theory! But then the new theory will generate a new relation $\textsc{prft}(\mathsf{m}, \mathsf{n})$ and a new $Prft(x, y)$ and so a new unprovable sentence $\mathscr{R}$. So any theory extending Q is negation incomplete.

But it is worth a word about what are theories extending Q. Any such theory should build in equivalents of the $\mathscr{L}_{NT}$ vocabulary $\emptyset$, S, $+$, and $\times$ — and should have a predicate $Nat(x)$ to identify a class of objects to count as the numbers. Then if the theory makes the axioms of Q true on these objects, it is incomplete. Straightforward extensions of Q are ones like PA which simply add to its axioms. But ordinary ZF set theory also falls into this category — for it is possible to define a class of sets, say, $\varnothing$, $\{\varnothing\}$, $\{\varnothing, \{\varnothing\}\}$, $\{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}\ldots$ where any n is the set of all the numbers prior to it, along with operations on sets which obey the axioms of Q.[3] It follows that ZF is negation incomplete. In contrast, the domain for the standard theory of real numbers has all the entities required to do arithmetic. However that theory does not have a predicate $Nat(x)$ to pick out the natural numbers, and cannot recapitulate the theory of natural numbers on any subclass of its domain. So our incompleteness theorem does not get a grip, and in fact the theory of real numbers is demonstrably complete. Observe, though, that it is a *weakness* in this theory of real numbers, its inability to specify a certain class that makes room for its completeness.[4]

E13.3. Demonstrate T13.6.

13.2 Gödel's Second Theorem: Overview

We turn now to Gödel's second incompleteness theorem on the unprovability of consistency. In order to separate the forest from the trees, we divide this discussion into four main parts. First, in this section, Gödel's second theorem is proved subject to

[3]For discussion, see any introduction to set theory, for example, Enderton, *Elements of Set Theory*, chapter 4.

[4]There are real numbers $\mathsf{0}$ and $\mathsf{1}$; so it is natural to identify the integers with $\mathsf{0}$, $\mathsf{0}+\mathsf{1}$, $\mathsf{0}+\mathsf{1}+\mathsf{1}$ and so forth. The difficulty is to define a property within the theory of real numbers that picks out just the members of this series, as we have been able to define infinite recursive properties in $\mathscr{L}_{NT}$. The completeness of the theory of real numbers was originally proved by Tarski, and is discussed in books on model theory, for example, Hodges *A Shorter Model Theory*, theorems 2.7.2 and 7.4.4.

three *derivability conditions*. Then we turn to the derivability conditions themselves. The first is easy. But the second and third require extended discussion. There is some background (section 13.3). Then discussion of the second and third conditions (section 13.4 and section 13.5). This completes the proof. We conclude with some reflections and consequences from our results (section 13.6). There are alternative approaches to the second theorem.[5] Our's is a straight-ahead development of the standard approach based on the derivability conditions. This is, surely, a natural place to start. Ordinary texts end their discussion of the second theorem with the initial demonstration from the derivability conditions, offering just some general perspective on the rest.[6] However, even if you decide to bypass the details, this general perspective will be enhanced if you have some object at which to "wave" as you pass them by.

For this discussion we switch to PA. The result is that that PA and its its extensions cannot prove their own consistency. The reason for this switch will become vivid in demonstration of the derivability conditions — as many arguments that would have been by induction are forced into the theory and so are by IN.

Main agument. We have seen that for recursively axiomatized theories there is a recursive relation $\textsc{prft}(\mathsf{m},\mathsf{n})$. Since it is recursive, in theories extending Q, this relation is captured by a corresponding $\mathit{Prft}(x, y)$. Let,

$$\mathit{Prvt}(y) =_{\text{def}} \exists x \mathit{Prft}(x, y)$$

So $\mathit{Prvt}(y)$ just when something numbers a proof of the formula numbered y — when the formula numbered by y is provable. Insofar as the quantifier is unbounded, there is no suggestion that there is a corresponding recursive relation — in fact, we have seen in T12.20 that no recursive relation numbers the theorems of Q. Let,

$$\mathit{Cont} =_{\text{def}} \sim\mathit{Prvt}(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$$

So Cont is true just in case there is no proof of $\overline{0} = \overline{1}$. There are different ways to express consistency but, for theories extending Q this does as well as any other. Suppose T extends Q. If T is inconsistent, then it proves anything; so $T \vdash \overline{0} = \overline{1}$. Suppose $T \vdash \overline{0} = \overline{1}$; since T extends Q, $T \vdash \overline{0} \neq \overline{1}$; so it proves a contradiction and

[5] For references see section 3 of Raatikainen, "Gödel's Incompleteness Theorems." See also, Tourlakis, *Lectures in Logic and Set Theory: I*.

[6] So, for example, "the details of this are long and tedious, and will not be discussed here" (George and Velleman, *Philosophies of Mathematics*, 201; compare Boolos, Burgess and Jeffrey, *Computability and Logic*, 234.

is inconsistent. So T is inconsistent iff $T \vdash \overline{0} = \overline{1}$; and, transposing, T is consistent iff $T \nvdash \overline{0} = \overline{1}$.

The second theorem is this simple result: Under certain conditions, if T is consistent, then $T \nvdash Cont$. If it is consistent, then T cannot prove its own consistency. Suppose the first theorem applies to T, and suppose we could show,

$$(**) \qquad T \vdash Cont \to \sim Prvt(\overline{\ulcorner \mathscr{G} \urcorner})$$

Then, given what has gone before, we could make the following very simple argument. Suppose T is a recursively axiomatized theory extending Q.

> By T13.3, $T \vdash \mathscr{G} \leftrightarrow \sim\exists x Prft(x, \overline{\ulcorner \mathscr{G} \urcorner})$, which is to say, $T \vdash \mathscr{G} \leftrightarrow \sim Prvt(\overline{\ulcorner \mathscr{G} \urcorner})$; from this and (**), $T \vdash Cont \to \mathscr{G}$; so if $T \vdash Cont$ then $T \vdash \mathscr{G}$; but from the first theorem (T13.4), if T is consistent, then $T \nvdash \mathscr{G}$; so if T is consistent, $T \nvdash Cont$.

So the argument reduces to showing (**). Observe that, in reasoning for T13.4 we have already shown,

$$T \text{ is consistent} \Longrightarrow T \nvdash \mathscr{G}$$

So the argument reduces to showing that T *proves* what we have already seen is so.

Let us abbreviate $Prvt(\overline{\ulcorner \mathscr{P} \urcorner})$ by $\Box\mathscr{P}$. Observe that this obscures the corner quotes. Still, we shall find it useful. So we need $T \vdash Cont \to \sim\Box\mathscr{G}$, which is just to say, $T \vdash \sim\Box(\overline{0} = \overline{1}) \to \sim\Box\mathscr{G}$. Suppose T satisfies the following *derivability conditions*.

D1. If $T \vdash \mathscr{P}$ then $T \vdash \Box\mathscr{P}$

D2. $T \vdash \Box(\mathscr{P} \to \mathscr{Q}) \to (\Box\mathscr{P} \to \Box\mathscr{Q})$

D3. $T \vdash \Box\mathscr{P} \to \Box\Box\mathscr{P}$

Then we shall be able to show $T \vdash Cont \to \sim\Box\mathscr{G}$.

The utility of $\Box$ in this context is that D1 - D3 are exactly the conditions that define a standard modal logic, K4 — and it is not surprising that *provability* should correspond to a kind of necessity.[7] There is an elegant natural derivation system for this modal logic. For this you might check out Roy, Natural Derivations for Priest §2

[7] While K4 correctly represents these principles, it is not a complete logic of provability. The complete system GL of provability for PA strengthens D3 to an axiom $\Box(\Box\mathscr{P} \to \mathscr{P}) \to \Box\mathscr{P}$. For discussion see Boolos, *The Logic of Provability*.

(but in the nomenclature there borrowed from Priest, the system is $NK\tau$). However rather than explain and introduce a new derivation system, we obtain a version of K4 simply by adding A1 - A3 and MP from AD to D1 - D3. So K4 has D1 as a new rule, and D2 and D3 as new axioms. Since A1 - A3 and MP remain, we have all the theorems from before. Thus, as a simple example,

(A)	1. $\sim\mathscr{P} \to (\mathscr{P} \to \mathscr{Q})$	T3.9
	2. $\Box[\sim\mathscr{P} \to (\mathscr{P} \to \mathscr{Q})]$	1 D1
	3. $\Box[\sim\mathscr{P} \to (\mathscr{P} \to \mathscr{Q})] \to [\Box\sim\mathscr{P} \to \Box(\mathscr{P} \to \mathscr{Q})]$	D2
	4. $\Box\sim\mathscr{P} \to \Box(\mathscr{P} \to \mathscr{Q})$	3,2 MP

So in this system $\vdash \Box\sim\mathscr{P} \to \Box(\mathscr{P} \to \mathscr{Q})$.

Now, given that $T \vdash \mathscr{G} \to \sim\exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$ from T13.3 we shall be able to show that $T \vdash Cont \to \sim\Box\mathscr{G}$.

T13.9. Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic of provability, $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$.

1.	$\mathscr{G} \to \sim\Box\mathscr{G}$	from T13.3
2.	$\Box(\mathscr{G} \to \sim\Box\mathscr{G})$	1 D1
3.	$\Box(\mathscr{G} \to \sim\Box\mathscr{G}) \to (\Box\mathscr{G} \to \Box\sim\Box\mathscr{G})$	D2
4.	$\Box\mathscr{G} \to \Box\sim\Box\mathscr{G}$	3,2 MP
5.	$\Box\sim\Box\mathscr{G} \to \Box(\Box\mathscr{G} \to \overline{0} = \overline{1})$	(A)
6.	$\Box\mathscr{G} \to \Box(\Box\mathscr{G} \to \overline{0} = \overline{1})$	4,5 T3.2
7.	$\Box(\Box\mathscr{G} \to \overline{0} = \overline{1}) \to (\Box\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))$	D2
8.	$\Box\mathscr{G} \to (\Box\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))$	6,7 T3.2
9.	$[\Box\mathscr{G} \to (\Box\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))] \to [(\Box\mathscr{G} \to \Box\Box\mathscr{G}) \to (\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))]$	A2
10.	$(\Box\mathscr{G} \to \Box\Box\mathscr{G}) \to (\Box\mathscr{G} \to \Box(\overline{0} = \overline{1}))$	9,8 MP
11.	$\Box\mathscr{G} \to \Box\Box\mathscr{G}$	D3
12.	$\Box\mathscr{G} \to \Box(\overline{0} = \overline{1})$	10,11 MP
13.	$[\Box\mathscr{G} \to \Box(\overline{0} = \overline{1})] \to [\sim\Box(\overline{0} = \overline{1}) \to \sim\Box\mathscr{G}]$	T3.13
14.	$\sim\Box(\overline{0} = \overline{1}) \to \sim\Box\mathscr{G}$	13,12 MP

Which is to say, $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$.

As usual for an axiomatic derivation, the reasoning is not entirely transparent. However we are at the stage where, given the derivability conditions, T proves the result. Given this, reason as before,

T13.10. Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions, if T is consistent, $T \nvdash Cont$.

Suppose T is a recursively axiomatized theory extending Q that satisfies the derivability conditions. Then by T13.9, $T \vdash \mathit{Cont} \rightarrow \sim\mathit{Prvt}(\overline{\ulcorner \mathscr{G} \urcorner})$; and by T13.3, $T \vdash \mathscr{G} \leftrightarrow \sim\mathit{Prvt}(\overline{\ulcorner \mathscr{G} \urcorner})$; so $T \vdash \mathit{Cont} \rightarrow \mathscr{G}$; so if $T \vdash \mathit{Cont}$ then $T \vdash \mathscr{G}$; but from the first incompleteness theorem (T13.4), if T is consistent, then $T \nvdash \mathscr{G}$; so if T is consistent, $T \nvdash \mathit{Cont}$.

One might wonder about the significance of this theorem: If T were inconsistent, it *would* prove *Cont*. So a failure to prove *Cont* is no reason to think that T is inconsistent. And a proof of *Cont* might itself be an indication of inconsistency! The interesting point here results from using one theory to prove the consistency of another. Recall the main Hilbert strategy as outlined in the introduction to Part IV; a key component is the demonstration by means of some real theory R that an ideal theory I is consistent. But, supposing that PA cannot prove its own consistency, we can be sure that no *weaker* theory can prove the consistecy of PA. And if PA cannot prove even the consistency of PA, then PA and theories weaker than PA cannot be used to prove the consistency of theories *stronger* than PA. So a leg of the Hilbert strategy seems to be removed. Observe, however, that the theorem does not show that the consistency of PA is unprovable: a theory stronger than PA at least in some respects might still prove the consistency of PA.[8] This may be a straightforward theorem of the second theory. Of course, as a means of demonstrating consistency such an argument may seem problematic insofar as one requires some reason for thinking the second theory sound which does not already attach to the first, and so already show that the first theory is consistent.

Another theorem is easy to show, and left as an exercise.

T13.11. Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic of provability, $T \vdash \mathit{Cont} \leftrightarrow \sim\mathit{Prvt}(\overline{\ulcorner \mathit{Cont} \urcorner})$.

Hints: (i) Show that $T \vdash \mathit{Cont} \rightarrow \sim\Box\mathit{Cont}$; you can do this starting with $\mathit{Cont} \rightarrow \sim\Box\mathscr{G}$ from T13.9 and $\sim\Box\mathscr{G} \rightarrow \mathscr{G}$ from T13.3. Then (ii) show $T \vdash \sim\Box\mathit{Cont} \rightarrow \mathit{Cont}$; for this, use T3.39 with T3.9 to show $T \vdash \overline{\mathsf{0}} = \overline{\mathsf{1}} \rightarrow \mathit{Cont}$; then you should be able to obtain $\sim\Box\mathit{Cont} \rightarrow \sim\Box(\overline{\mathsf{0}} = \overline{\mathsf{1}})$ which is to say $\sim\Box\mathit{Cont} \rightarrow \mathit{Cont}$. Together these give the desired result.

[8] G. Gentzen shows this very thing, "The Consistency of Elementary Number Theory," and "New Version of the Consistency Proof for Elementary Number Theory," both in *The Collected Papers of Gehard Gentzen*, ed. Szabo. See also Gentzen, "The Concept of Infinite in Mathematics" also in Szabo, along with Pohlers, *Proof Theory*, chpater 1, and Takeuti, *Proof Theory*, §12.

From this theorem, supposing the derivability conditions, *Cont* is another $\mathcal{P}$ which, like $\mathcal{G}$, is such that $T \vdash \mathcal{P} \leftrightarrow \sim Prvt(\overline{\ulcorner \mathcal{P} \urcorner})$; so *Cont* is another fixed point for $\sim Prvt(x)$. It follows that *Cont* is another sentence such that both it and its negation are unprovable. Interestingly, *Cont* uses the notion of provability, but is not constructed so as to say anything about its *own* provability — and so this instance of incompleteness does not depend on self-reference for the unprovable sentence.

We have shown that the second theorem holds for a theory if it meets the derivability conditions. But this is not to show that the theorem holds for any theories! In order to tie the result to something concrete, we turn now to showing that PA meets the derivability conditions, and so that PA, and theories extending PA, satisfy the theorem.

Demonstration of the first condition is simple.

T13.12. Suppose T is a recursively axiomatized theory extending Q. Then if $T \vdash \mathcal{P}$, then $T \vdash \Box\mathcal{P}$.

Suppose $T \vdash \mathcal{P}$; then since T is recursively axiomatized, for some m, $\textsc{prft}(\mathsf{m}, \ulcorner \mathcal{P} \urcorner)$; and since T extends Q, there is a *Prft* that captures $\textsc{prft}$; so $T \vdash Prft(\overline{\mathsf{m}}, \overline{\ulcorner \mathcal{P} \urcorner})$; so by $\exists$I, $T \vdash \exists x Prft(x, \overline{\ulcorner \mathcal{P} \urcorner})$; so $T \vdash Prvt(\overline{\ulcorner \mathcal{P} \urcorner})$; so $T \vdash \Box\mathcal{P}$.

The next conditions are considerably more difficult. We build gradually to the required results in PA.

E13.4. Produce derivations to show both parts of T13.11.

13.3 The Derivability Conditions: Background

In this section we develop some results required for demonstration of derivability conditions two and three. We proceed by introducing functions and relations into PA by *definition*, and then proving some results about them.

13.3.1 Remarks on Definition

To obtain the derivability conditions, we begin with some remarks on definition. So far, we have taken a language, as $\mathcal{L}_q$ or $\mathcal{L}_{NT}$ as basic, and introduced any additional symbols, for example $\leq$, as means of *abbreviation* for expressions in the original language. But in more complex contexts — especially involving function symbols, it

Additional Theorems of PA

*T13.13. *The following are theorems of PA*:

(a) PA $\vdash (r \leq s \wedge s \leq t) \rightarrow r \leq t$

(b) PA $\vdash (r < s \wedge s < t) \rightarrow r < t$

(c) PA $\vdash (r \leq s \wedge s < t) \rightarrow r < t$

(d) PA $\vdash \emptyset \leq t$

(e) PA $\vdash \emptyset < St$

(f) PA $\vdash t \neq \emptyset \leftrightarrow \emptyset < t$

(g) PA $\vdash t < St$

(h) PA $\vdash St = s \rightarrow t < s$

(i) PA $\vdash s \leq t \leftrightarrow Ss \leq St$

(j) PA $\vdash s < t \leftrightarrow Ss < St$

(k) PA $\vdash s < t \leftrightarrow Ss \leq t$

(l) PA $\vdash s \leq t \leftrightarrow s < t \vee s = t$

(m) PA $\vdash s < St \leftrightarrow s < t \vee s = t$

(n) PA $\vdash s \leq St \leftrightarrow s \leq t \vee s = St$

(o) PA $\vdash s < t \vee s = t \vee t < s$

(p) PA $\vdash s \leq t \vee t < s$

(q) PA $\vdash s \leq t \leftrightarrow t \not< s$

(r) PA $\vdash t < s \rightarrow t \neq s$

(s) PA $\vdash (s \leq t \wedge t \leq s) \rightarrow s = t$

(t) PA $\vdash s \leq s + t$

(u) PA $\vdash r \leq s \leftrightarrow r + t \leq s + t$

(v) PA $\vdash r < s \leftrightarrow r + t < s + t$

(w) PA $\vdash (r \leq s \wedge t \leq u) \rightarrow r + t \leq s + u$

(x) PA $\vdash (r < s \wedge t \leq u) \rightarrow r + t < s + u$

(y) PA $\vdash \emptyset < t \rightarrow s \leq s \times t$

(z) PA $\vdash r \leq s \rightarrow r \times t \leq s \times t$

(aa) PA $\vdash r \times s > \emptyset \rightarrow s > \emptyset$

(ab) PA $\vdash (r > \overline{1} \wedge s > \emptyset) \rightarrow r \times s > s$

(ac) PA $\vdash (t > \emptyset \wedge r < s) \rightarrow r \times t < s \times t$

(ad) PA $\vdash (r < s \wedge t < u) \rightarrow r \times t < s \times u$

(ae) PA $\vdash \forall x[(\forall z < x)\mathcal{P}^x_z \rightarrow \mathcal{P}] \rightarrow \forall x\mathcal{P}$ *strong induction (a)*

(af) PA $\vdash \mathcal{P}^x_\emptyset \wedge \forall x[(\forall z \leq x)\mathcal{P}^x_z \rightarrow \mathcal{P}^x_{Sx}] \rightarrow \forall x\mathcal{P}$ *strong induction (b)*

(ag) PA $\vdash \exists x\mathcal{P} \rightarrow \exists x[\mathcal{P} \wedge (\forall z < x)\sim\mathcal{P}^x_z]$ *least number principle*

Some of these are related to results we obtained in chapter 8 for Q. But there results were of the sort, for any n, Q $\vdash t < \overline{\mathsf{n}} \vee t = \overline{\mathsf{n}} \vee \overline{\mathsf{n}} < t$; with PA, the induction is in the logic rather than in the metalanguage, and we obtain the universal quantifier (or rather, an arbitrary term which may be a free variable) in the object formula.

will be convenient to *extend* the language with the addition of new symbols by means of definition. Thus given a theory T in language $\mathcal{L}$, we might introduce symbols with corresponding axioms to obtain T' and $\mathcal{L}'$ as follows,

Symbol	*Axiom*	*Condition*
$\exists$	$\exists x\mathcal{P} \leftrightarrow \sim\forall x\sim\mathcal{P}$	
$\leq$	$x \leq y \leftrightarrow \exists z(z + x = y)$	
$\emptyset$	$y = \emptyset \leftrightarrow \forall x(x \notin y)$	$T \vdash \exists! y\forall x(x \notin y)$
S	$y = Sx \leftrightarrow \forall z[z \in y \leftrightarrow (z \in x \vee z = x)]$	$T \vdash \exists! y\forall z[z \in y \leftrightarrow (z \in x \vee z = x)]$

We are familiar with the first two cases, for an operator and a relation symbol. Strictly, the first is an axiom schema, representing different axioms for different instances of $\mathcal{P}$. So far, we have thought of these as *abbreviations* — and as such the listed axioms are of the sort $\mathcal{Q} \leftrightarrow \mathcal{Q}$ with the abbreviated form on one side, and the unabbreviated on the other. A theory is not extended by the addition of an "axiom" of this sort. We will continue to see the introduction of operators this way. It is simplest to think of relation symbols as abbreviations too. However, we shall also be able to see them as (relatively easy) examples of *new* vocabulary — and they are introduced below in this way. For the others, let $\exists! y\mathcal{P}(y)$ abbreviate $\exists y[\mathcal{P}(y) \wedge \forall z(\mathcal{P}(z) \rightarrow z = y)]$ or equivalently $\exists y\mathcal{P}(y) \wedge \forall y\forall z[(\mathcal{P}(y) \wedge \mathcal{P}(z)) \rightarrow y = z]$ so that *exactly one* thing is $\mathcal{P}$. Then the cases for a constant and function symbol are standard examples from set theory, where zero and successor are defined (the condition for successor sets $Sx = x \cup \{x\}$ so that the integers are $\emptyset$, $\{\emptyset\}$, $\{\emptyset, \{\emptyset\}\}$ and so forth). Observe that the constant and function cases require that T prove a uniqueness condition for the symbol. The details of the examples are not important; we illustrate only the idea of definition. We begin with a formal account, and extend it in different directions.

Basic Account

Consider some theory T and language $\mathcal{L}$. We will consider a language $\mathcal{L}'$ extended with some new symbol and theory T' extended with the corresponding axiom. There are separate cases for a relation symbol, constant and function symbol.

Relation symbol. To introduce a new relation symbol $\mathcal{R}\vec{x}$ we require an axiom in the extended theory such that,

$$T' \vdash \mathcal{R}(\vec{x}) \leftrightarrow \mathcal{Q}(\vec{x})$$

where $\mathcal{Q}(\vec{x})$ is in $\mathcal{L}$. Then for a formula $\mathcal{F}'$ including the new symbol, there should be a conversion $\mathfrak{C}$ such that $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}$ for $\mathcal{F}$ in the original $\mathcal{L}$ and,

$$T' \vdash \mathcal{F}' \qquad \text{iff} \qquad T \vdash \mathfrak{C}[\mathcal{F}']$$

So $\mathcal{F}$ is like our unabbreviated formula, always available in the original T when $\mathcal{F}'$ is a theorem of T'. The conversion for a relation $\mathcal{R}\vec{\mathfrak{s}}$ is straightforward. We are given $T \vdash \mathcal{R}(\vec{x}) \leftrightarrow \mathcal{Q}(\vec{x})$. Make sure the bound variables of $\mathcal{Q}$ do not overlap the variables of $\mathcal{R}\vec{\mathfrak{s}}$. Then $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}'^{\mathcal{R}\vec{\mathfrak{s}}}_{\mathcal{Q}(\vec{\mathfrak{s}})}$. So, from the example above, suppose $\mathcal{F}' = \exists z(a \leq z)$. So $\mathcal{F}'$ involves the new symbol. We are given $T' \vdash x \leq y \leftrightarrow \exists z(z + x = y)$ so that $\mathcal{R}(\vec{x}) = x \leq y$ and $\mathcal{Q}(\vec{x}) = \exists z(z+x = y)$. It will not make sense to instantiate x and y from this $\mathcal{Q}$ to a and z from $\mathcal{F}'$ insofar as z is not free in $\mathcal{Q}$. But we solve the problem by revising bound variables; so $T' \vdash x \leq y \leftrightarrow \exists w(w + x = y)$; so $T' \vdash a \leq z \leftrightarrow \exists w(w + a = z)$; then $\mathfrak{C}[\mathcal{F}']$ replaces $(a \leq z)$ in $\mathcal{F}'$ with $\exists w(w + a = z)$ to obtain, $\exists z \exists w(w + a = z)$.

Constant symbol. To introduce a new constant symbol we require an axiom in the extended theory, along with a condition in the original theory such that,

$$T' \vdash y = c \leftrightarrow \mathcal{Q}(y) \qquad \text{and} \qquad T \vdash \exists! y \mathcal{Q}(y)$$

Again for a formula $\mathcal{F}'$ including the new symbol, we expect a conversion $\mathfrak{C}$ such that $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}$, where $T' \vdash \mathcal{F}'$ iff $T \vdash \mathfrak{C}[\mathcal{F}]$. Let z be a variable that does not appear in $\mathcal{F}'$ or $\mathcal{Q}$. Then

$$\mathfrak{C}[\mathcal{F}'] = \exists z(\mathcal{Q}(z) \wedge \mathcal{F}'^{c}_{z})$$

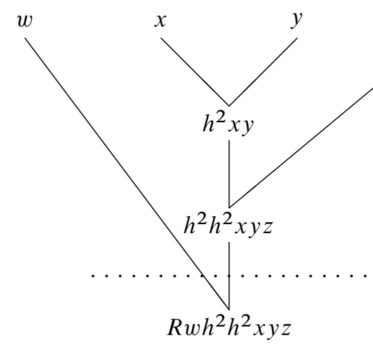
So, from the example above, suppose $\mathcal{F}' = \exists x(\emptyset \in x)$. Then z is a variable that does not appear in $\mathcal{F}'$ or $\mathcal{Q}$. So $\mathfrak{C}[\mathcal{F}'] = \exists z[\forall x(x \notin z) \wedge \exists x(z \in x)]$.

Function symbol. To introduce a function symbol, there is an axiom and condition,

$$T' \vdash y = h\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y) \qquad \text{and} \qquad T \vdash \exists! y \mathcal{Q}(\vec{x}, y)$$

The conversion for a function symbol works like that for constants when a single instance of $h\vec{\mathfrak{s}}$ appears in $\mathcal{F}'$. Again, make sure the bound variables of $\mathcal{Q}$ do not overlap the variables of $h\vec{\mathfrak{s}}$ and let z be a variable that does not appear in $\mathcal{F}'$ or in $\mathcal{Q}$. Then it is sufficient to set $\mathfrak{C}[\mathcal{F}'] = \exists z(\mathcal{Q}(\vec{\mathfrak{s}}, z) \wedge \mathcal{F}'^{h\vec{\mathfrak{s}}}_{z})$. In general, however, $\mathcal{F}'$ may include multiple instances of h, including one in the scope of another. For the general case, begin where $\mathcal{F}'$ is an atomic $\mathcal{R}' = \mathcal{R}t_1 \ldots t_n$ and $t_1 \ldots t_n$ may involve instances of $h\vec{\mathfrak{s}}$. Order instances of $h\vec{\mathfrak{s}}$ in $\mathcal{R}'$ from the left (or, on a chapter 2 tree, from the bottom) into a list $h\vec{\mathfrak{s}}_1, h\vec{\mathfrak{s}}_2, \ldots h\vec{\mathfrak{s}}_m$, so that when $i < j$, no $h\vec{\mathfrak{s}}_i$

appears in the scope of $\mathscr{h}\vec{\mathfrak{s}}_j$. Then set $\mathcal{R}_0 = \mathcal{R}'$, and for $i \geq 1$, $\mathcal{R}_i = \exists z(\mathcal{Q}(\vec{\mathfrak{s}}_i, z) \wedge (\mathcal{R}_{i-1})^{\mathscr{h}\vec{\mathfrak{s}}_i}_z)$. Then $\mathfrak{C}[\mathcal{R}'] = \mathcal{R}_m$ and for an arbitrary $\mathcal{F}'$, $\mathfrak{C}[\mathcal{F}'] = \mathcal{F}'^{\mathcal{R}'}_{\mathcal{R}_m}$. So, for example, if $\mathcal{R}' = \mathcal{R}_0 = Rwh^2h^2xyz$, the tree is as follows,

(B) 

So instances of $\mathscr{h}\mathscr{q}\mathscr{r}$ are ordered $\langle h^2h^2xyz, h^2xy\rangle$. Then we use $\mathcal{Q}$ to replace instances of $\mathscr{h}$, working our way up through the tree. So,

$$\mathcal{R}_0 = Rwh^2h^2xyz$$

$$\mathcal{R}_1 = \exists u(\mathcal{Q}h^2xyzu \wedge Rwu)$$

$$\mathcal{R}_2 = \exists v[\mathcal{Q}xyv \wedge \exists u(\mathcal{Q}vzu \wedge Rwu)]$$

$\mathcal{R}_1$ uses $\mathcal{Q}$ to replace all of h^2h^2xyz, operating on the terms h^2xy and z. $\mathcal{R}_2$ uses $\mathcal{Q}$ to replace h^2xy in $\mathcal{R}_1$, operating on x and y. Observe that no quantifier ever binds a variable still in the scope of h; and in the end, the free variables are the same as in $\mathcal{R}'$.

To show that this works, that $T' \vdash \mathcal{F}'$ iff $T \vdash \mathcal{F}$ we need a couple of theorems. The idea is to show that $T' \vdash \mathcal{F}' \leftrightarrow \mathcal{F}$ and then that $T' \vdash \mathcal{F}$ iff $T \vdash \mathcal{F}$. Together, these give the result we want. First,

T13.14. For a defined relation symbol, function symbol or constant, with its associated axiom and conversion procedure, $T' \vdash \mathcal{F}' \leftrightarrow \mathcal{F}$.

(a) For a relation symbol, we are given $T' \vdash \mathcal{R}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x})$; then so long as the bound variables of $\mathcal{Q}$ do not overlap the variables of $\mathcal{R}\vec{\mathfrak{s}}$ (which we can guarantee by reasoning as for T3.27) $\vec{\mathfrak{s}}$ is free for $\vec{x}$ in $\mathcal{Q}$, so $T' \vdash \mathcal{R}\vec{\mathfrak{s}} \leftrightarrow \mathcal{Q}(\vec{\mathfrak{s}})$; so by T9.9, $T' \vdash \mathcal{F}' \leftrightarrow \mathcal{F}'^{\mathcal{R}\vec{\mathfrak{s}}}_{\mathcal{Q}(\vec{\mathfrak{s}})}$; so $T' \vdash \mathcal{F}' \leftrightarrow \mathcal{F}$.

(b) The case for constants is left as an exercise.

(c) For a function symbol $\mathscr{h}$, begin with a derivation to show $T' \vdash \mathscr{R}_{i-1} \leftrightarrow \mathscr{R}_i$. For a $\mathscr{R}_{i-1}[\mathscr{h}(\vec{\mathfrak{z}})]$, $\mathscr{R}_i(\vec{\mathfrak{z}})$ is $\exists z(\mathscr{Q}(\vec{\mathfrak{z}}, z) \wedge \mathscr{R}_{i-1}[z])$. We have as an axiom that $T' \vdash y = \mathscr{h}\vec{x} \leftrightarrow \mathscr{Q}(\vec{x}, y)$

1.	$\mathscr{R}_{i-1}[\mathscr{h}(\vec{\mathfrak{z}})]$	A (g $\leftrightarrow$I)
2.	$\mathscr{h}(\vec{\mathfrak{z}}) = \mathscr{h}(\vec{\mathfrak{z}}) \leftrightarrow \mathscr{Q}(\vec{\mathfrak{z}}, \mathscr{h}(\vec{\mathfrak{z}}))$	from T'
3.	$\mathscr{h}\vec{\mathfrak{z}} = \mathscr{h}\vec{\mathfrak{z}}$	=I
4.	$\mathscr{Q}(\vec{\mathfrak{z}}, \mathscr{h}(\vec{\mathfrak{z}}))$	2,3 $\leftrightarrow$E
5.	$\mathscr{Q}(\vec{\mathfrak{z}}, \mathscr{h}(\vec{\mathfrak{z}})) \wedge \mathscr{R}_{i-1}[\mathscr{h}(\vec{\mathfrak{z}})]$	1,4 $\wedge$I
6.	$\exists z(\mathscr{Q}(\vec{\mathfrak{z}}, z) \wedge \mathscr{R}_{i-1}[z])$	5 $\exists$I
7.	$\exists z(\mathscr{Q}(\vec{\mathfrak{z}}, z) \wedge \mathscr{R}_{i-1}[z])$	A (g $\leftrightarrow$I)
8.	$\mathscr{Q}(\vec{\mathfrak{z}}, j) \wedge \mathscr{R}_{i-1}[j]$	A (g 7$\exists$E)
9.	$\mathscr{Q}(\vec{\mathfrak{z}}, j)$	8 $\wedge$E
10.	$j = \mathscr{h}(\vec{\mathfrak{z}}) \leftrightarrow \mathscr{Q}(\vec{\mathfrak{z}}, j)$	from T'
11.	$j = \mathscr{h}(\vec{\mathfrak{z}})$	10,9 $\leftrightarrow$E
12.	$\mathscr{R}_{i-1}[j]$	8 $\wedge$E
13.	$\mathscr{R}_{i-1}[\mathscr{h}(\vec{\mathfrak{z}})]$	11,12 =E
14.	$\mathscr{R}_{i-1}[\mathscr{h}(\vec{\mathfrak{z}})]$	7,8-13 $\exists$E
15.	$\mathscr{R}_{i-1}[\mathscr{h}(\vec{\mathfrak{z}})] \leftrightarrow \exists z(\mathscr{Q}(\vec{\mathfrak{z}}, z) \wedge \mathscr{R}_{i-1}[z])$	1-6,7-14 $\leftrightarrow$I

Things are arranged so that the variables of $\mathscr{h}\vec{\mathfrak{z}}$ are not bound upon substitution into $\mathscr{Q}$. So instances of the axiom at (2) and (10) and $\exists$I at (6) satisfy constraints. So $T' \vdash \mathscr{R}_{i-1} \leftrightarrow \mathscr{R}_i$; and by repeated applications of this theorem, $T' \vdash \mathscr{R}' \leftrightarrow \mathscr{R}_m$; so by T9.9, $T' \vdash \mathscr{F}' \leftrightarrow \mathscr{F}'^{\mathscr{R}'}_{\mathscr{R}_m}$; so $T' \vdash \mathscr{F}' \leftrightarrow \mathscr{F}$.

So far, so good, but this only says what the extended T' proves — that the richer T' proves $\mathscr{F}'$ iff it proves $\mathscr{F}$. But we want to see that T' proves $\mathscr{F}'$ iff the original T proves $\mathscr{F}$. We bridge the gap between T and T' by an additional theorem.

T13.15. For a T and $\mathscr{L}$, given a defined relation symbol, function symbol or constant with its associated axiom, then for any formula $\mathscr{F}$ in the original $\mathscr{L}$, $T' \vdash \mathscr{F}$ iff $T \vdash \mathscr{F}$.

Since T' proves everything T proves, the direction from right to left is obvious. So suppose $T' \vdash \mathscr{F}$. To show $T \vdash \mathscr{F}$, we show $T \vDash \mathscr{F}$ and apply adequacy. So suppose there is a model M such that $\mathsf{M}[T] = \mathsf{T}$; our aim is to show $\mathsf{M}[\mathscr{F}] = \mathsf{T}$. Since $T' \vdash \mathscr{F}$, by soundness, $T' \vDash \mathscr{F}$.

(i) Relation symbol. Extend M to a model M' like M except that for arbitrary d, $\langle \mathsf{d}[x_1] \ldots \mathsf{d}[x_n] \rangle \in \mathsf{M}'[\mathscr{R}]$ iff $\mathsf{M}_{\mathsf{d}}[\mathscr{Q}(x_1 \ldots x_n)] = \mathsf{S}$; iff $\mathsf{M}'_{\mathsf{d}}[\mathscr{Q}(x_1 \ldots x_n)] = \mathsf{S}$ (the latter by T10.15 since M and M' agree on assignments to symbols

in $\mathcal{Q}$). Since M' and M agree on assignments to symbols other than $\mathcal{R}$, by T10.15 $\mathsf{M}'[T] = \mathsf{T}$. And $\mathsf{M}'[\mathcal{R}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x})] = \mathsf{T}$: suppose otherwise; then by TI there is some d such that $\mathsf{M}'_{\mathsf{d}}[\mathcal{R}x_1 \ldots x_n \leftrightarrow \mathcal{Q}(x_1 \ldots x_n)] \neq \mathsf{S}$; so by SF($\leftrightarrow$), $\mathsf{M}'_{\mathsf{d}}[\mathcal{R}x_1 \ldots x_n] \neq \mathsf{S}$ and $\mathsf{M}'_{\mathsf{d}}[\mathcal{Q}(x_1 \ldots x_n)] = \mathsf{S}$ (or the other way around); so $\langle \mathsf{d}[x_1] \ldots \mathsf{d}[x_n] \rangle \notin \mathsf{M}'[\mathcal{R}]$ and $\mathsf{M}'_{\mathsf{d}}[\mathcal{Q}(x_1 \ldots x_n)] = \mathsf{S}$; but by construction, this is impossible; and similarly in the other case; reject the assumption, $\mathsf{M}'[\mathcal{R}\vec{x} \leftrightarrow \mathcal{Q}(\vec{x})] = \mathsf{T}$. So $\mathsf{M}'[T'] = \mathsf{T}$; so since $T' \vDash \mathcal{F}$, $\mathsf{M}'[\mathcal{F}] = \mathsf{T}$; and by T10.15 again, $\mathsf{M}[\mathcal{F}] = \mathsf{T}$; and since this reasoning applies for arbitrary M, $T \vDash \mathcal{F}$; so by adequacy, $T \vdash \mathcal{F}$.

(ii) Again, the case for constants is left as an exercise.

(iii) Function symbol. Since $T \vdash \exists! y\mathcal{Q}(\vec{x}, y)$, by soundness $T \vDash \exists! y\mathcal{Q}(\vec{x}, y)$; so since $\mathsf{M}[T] = \mathsf{T}$, $\mathsf{M}[\exists! y\mathcal{Q}(\vec{x}, y)] = \mathsf{T}$; so by TI, for any d, $\mathsf{M}_{\mathsf{d}}[\exists! y\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$, and there is exactly one $\mathsf{m} \in \mathsf{U}$ such that $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$. Extend M to a model M' like M except that for arbitrary d, $\langle\langle \mathsf{d}[x_1] \ldots \mathsf{d}[x_n]\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\hbar]$ iff $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(x_1 \ldots x_n, y)] = \mathsf{S}$; by T10.15 iff $\mathsf{M}'_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(x_1 \ldots x_n, y)] = \mathsf{S}$. Since M' and M agree on assignments to symbols other than $\hbar$, by T10.15 $\mathsf{M}'[T] = \mathsf{T}$. And $\mathsf{M}'[y = \hbar\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y)] = \mathsf{T}$: suppose otherwise; then by TI there is some h such that $\mathsf{M}'_{\mathsf{h}}[y = \hbar x_1 \ldots x_n \leftrightarrow \mathcal{Q}(x_1 \ldots x_n, y)] \neq \mathsf{S}$; so by SF($\leftrightarrow$), $\mathsf{M}'_{\mathsf{h}}[y = \hbar x_1 \ldots x_n] \neq \mathsf{S}$ and $\mathsf{M}'_{\mathsf{h}}[\mathcal{Q}(x_1 \ldots x_n, y)] = \mathsf{S}$ (or the other way around). Where $\mathsf{h}(y) = \mathsf{m}$, $\mathsf{h} = \mathsf{h}(y|\mathsf{m})$, and $\mathsf{M}'_{\mathsf{h}(y|\mathsf{m})}[\mathcal{Q}(x_1 \ldots x_n, y)] = \mathsf{S}$; so by construction with TA(f), $\mathsf{M}'_{\mathsf{h}}[\hbar x_1 \ldots x_n] = \mathsf{m}$; and since $\mathsf{h}(y) = \mathsf{m}$, $\mathsf{M}'_{\mathsf{h}}[y] = \mathsf{m}$; so $\mathsf{M}'_{\mathsf{h}}[y = \hbar x_1 \ldots x_n] = \mathsf{S}$; this is impossible; and similarly in the other case; reject the assumption, $\mathsf{M}'[y = \hbar\vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, y)] = \mathsf{T}$. So $\mathsf{M}'[T'] = \mathsf{T}$; so since $T' \vDash \mathcal{F}$, $\mathsf{M}'[\mathcal{F}] = \mathsf{T}$; and by T10.15 again, $\mathsf{M}[\mathcal{F}] = \mathsf{T}$; and since this reasoning applies for arbitrary M, $T \vDash \mathcal{F}$; so by adequacy, $T \vdash \mathcal{F}$.

It is, in fact, important to show that these specifications are consistent — that we do not both assert and deny that some objects are in the interpretation of a relation symbol, function symbol or constant when we specify for assignments that are arbitrary. But this is easily done. Here is the case for function symbols.

This specification is consistent: Suppose otherwise; that is, suppose there are some assignments d and h such that $\langle\langle \mathsf{d}[x_1] \ldots \mathsf{d}[x_n]\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\hbar]$ and $\langle\langle \mathsf{h}[x_1] \ldots \mathsf{h}[x_n]\rangle, \mathsf{m}\rangle \notin \mathsf{M}'[\hbar]$ but $\mathsf{d}[x_1] = \mathsf{h}[x_1]$ and ... and $\mathsf{d}[x_n] = \mathsf{h}[x_n]$. From the first, $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(x_1 \ldots x_n, y)] = \mathsf{S}$; from the second, $\mathsf{M}_{\mathsf{h}(y|\mathsf{m})}[\mathcal{Q}(x_1 \ldots x_n, y)] \neq \mathsf{S}$; but $\mathsf{d}(\mathsf{y}|\mathsf{m})$ and $\mathsf{h}(\mathsf{y}|\mathsf{m})$ make the same assignments to variables free in $\mathcal{Q}(\vec{x}, y)$; so by T8.4, $\mathsf{M}_{\mathsf{d}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{M}_{\mathsf{h}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)]$; so

$\mathsf{M}_{\mathsf{h}(y|\mathsf{m})}[\mathcal{Q}(\vec{x}, y)] = \mathsf{S}$; reject the assumption: if $\mathsf{d}[x_1] = \mathsf{h}[x_1]$ and ... and $\mathsf{d}[x_n] = \mathsf{h}[x_n]$ and $\langle\langle \mathsf{d}[x_1] \ldots \mathsf{d}[x_n]\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\hbar]$ then $\langle\langle \mathsf{h}[x_1] \ldots \mathsf{h}[x_n]\rangle, \mathsf{m}\rangle \in \mathsf{M}'[\hbar]$.

And now our desired result is simple. The basic idea is that for some T and $\mathcal{L}$ with a defined constant, relation symbol or function symbol, from T13.14 $T' \vdash \mathcal{F}' \leftrightarrow \mathcal{F}$ and from T13.15 $T' \vdash \mathcal{F}$ iff $T \vdash \mathcal{F}$; so that $T' \vdash \mathcal{F}'$ iff $T \vdash \mathcal{F}$. Put more generally,

T13.16. For some defined relation symbols, function symbols or constants, with their associated axioms and conversion procedures, $T' \vdash \mathcal{F}'$ iff $T \vdash \mathcal{F}$.

Consider a sequence of formulas $\mathcal{F}_0 \ldots \mathcal{F}_n$ and theories $T_0 \ldots T_n$ ordered according to the number of new symbols where for any i, $\mathcal{F}_i = \mathfrak{C}[\mathcal{F}_{i+1}]$. By our results, $T_{i+1} \vdash \mathcal{F}_{i+1} \leftrightarrow \mathcal{F}_i$, and $T_{i+1} \vdash \mathcal{F}_i$ iff $T_i \vdash \mathcal{F}_i$. It follows that $T_{i+1} \vdash \mathcal{F}_{i+1}$ iff $T_i \vdash \mathcal{F}_i$. And by a simple induction, $T_n \vdash \mathcal{F}_n$ iff $T_0 \vdash \mathcal{F}_0$, which is to say $T' \vdash \mathcal{F}'$ iff $T \vdash \mathcal{F}$.

In the following, we will be clear about when new symbols and associated axioms are introduced, and about the conditions under which this may be done. In light of the results we have achieved however, we will not generally distinguish between a theory and its definitional extensions.

It is worth remarking on the increased requirement for definition relative to capture. In particular, for a function, capture requires $T \vdash \forall z[\mathcal{F}(\overline{\mathsf{m}}_1 \ldots \overline{\mathsf{m}}_\mathsf{n}, z) \rightarrow z = \overline{\mathsf{a}}]$. For definition, from uniqueness, the comparable condition is $T \vdash \forall y \forall z[(\mathcal{F}(\vec{x}, y) \wedge \mathcal{F}(\vec{x}, z)) \rightarrow y = z]$. So definition builds in a sort of generality not required in the other case. Q is great about proving particular facts — but not so great when it comes to generality (this was a sticking point about the shift between Q and $\mathrm{Q_s}$ in chapter 12 (p. 570 and below). But this is just the sort of thing PA is fitted to do.[9]

*E13.5. Show T13.13ae and T13.13ag. Hard core: show each of the results in T13.13.

E13.6. (i) Complete the unfinished cases for constants in T13.14 and T13.15. (ii) Show consistency results for both relation and constant symbols.

[9] Is definition so described *necessary* for reasoning to follow? We might continue to think in terms of abbreviation — or even unabbreviated formulas themselves, so that there are no *new* symbols. Even so, the conditions on such formulas would be like those for definition, so that the overall argument would remain the same.

First applications

Here are a couple of quick results that will be helpful as we move forward. First, if PA defines some functions $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$, then PA defines their composition, $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$. We introduce a definition and then show that the condition is met. This pattern will repeat many times.

T13.17. If PA defines some $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$, then PA defines $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$. Suppose PA defines some $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$. Let,

Def$[f(\vec{x}, \vec{y}, \vec{z})]$ PA $\vdash v = f(\vec{x}, \vec{y}, \vec{z}) \leftrightarrow v = h(\vec{x}, g(\vec{y}), \vec{z})$. Then,

(i) PA $\vdash \exists v[v = h(\vec{x}, g(\vec{y}), \vec{z})]$

1.	$h(\vec{x}, g(\vec{y}), \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$	=I
2.	$\exists v[v = h(\vec{x}, g(\vec{y}), \vec{z})]$	1 ∃I

(ii) PA $\vdash \forall u \forall v[(u = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge v = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow u = v]$

1.	$j = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge k = h(\vec{x}, g(\vec{y}), \vec{z})$	A (g →I)
2.	$j = h(\vec{x}, g(\vec{y}), \vec{z})$	1 ∧E
3.	$k = h(\vec{x}, g(\vec{y}), \vec{z})$	1 ∧E
4.	$j = k$	2,3 =E
5.	$(j = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge k = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow j = k$	1-4 →I
6.	$\forall v[(j = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge v = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow j = v]$	5 ∀I
7.	$\forall u \forall v[(u = h(\vec{x}, g(\vec{y}), \vec{z}) \wedge v = h(\vec{x}, g(\vec{y}), \vec{z})) \rightarrow u = v]$	6 ∀I

So PA $\vdash \exists! v[v = h(\vec{x}, g(\vec{y}), \vec{z})]$ and PA defines $f(\vec{x}, \vec{y}, \vec{z})$.

In addition, we can introduce a function for *minimization*. The idea is to set $v = \mu y \mathcal{Q}(\vec{x}, y) \leftrightarrow [\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) \sim \mathcal{Q}(\vec{x}, z)]$. In the ordinary case, a new function symbol $\hbar$ is introduced with an axiom of the sort $v = \hbar \vec{x} \leftrightarrow \mathcal{Q}(\vec{x}, v)$ under the condition $T \vdash \exists! v \mathcal{Q}(\vec{x}, v)$. But, in this case, the situation is simplified by the following theorem.

T13.18. If PA $\vdash \exists v \mathcal{Q}(\vec{x}, v)$, then PA $\vdash \exists! v[\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) \sim \mathcal{Q}(\vec{x}, v)]$.

(i) Suppose PA $\vdash \exists v \mathcal{Q}(\vec{x}, v)$. Then by the least number principle T13.13ag, PA $\vdash \exists v[\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) \sim \mathcal{Q}(\vec{x}, v)]$.

(ii) Further, PA $\vdash \forall u \forall v[(\mathcal{Q}(\vec{x}, u) \wedge (\forall z < u) \sim \mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) \sim \mathcal{Q}(\vec{x}, z)) \rightarrow u = v]$.

1.	$\mathcal{Q}(\vec{x}, j) \wedge (\forall z < j)\sim\mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, k) \wedge (\forall z < k)\sim\mathcal{Q}(\vec{x}, z)$	A (g →I)
2.	$j < k \vee j = k \vee k < j$	T13.13o
3.	$j < k$	A (c ∼I)
4.	$(\forall z < k)\sim\mathcal{Q}(\vec{x}, z)$	1 ∧E
5.	$\sim\mathcal{Q}(\vec{x}, j)$	4,3 (∀E)
6.	$\mathcal{Q}(\vec{x}, j)$	1 ∧E
7.	$\bot$	6,5 ⊥I
8.	$\sim(j < k)$	3-7 ∼I
9.	$k < j$	A (c ∼I)
10.	$(\forall z < j)\sim\mathcal{Q}(\vec{x}, z)$	1 ∧E
11.	$\sim\mathcal{Q}(\vec{x}, k)$	10,9 (∀E)
12.	$\mathcal{Q}(\vec{x}, k)$	1 ∧E
13.	$\bot$	12,11, ⊥I
14.	$\sim(k < j)$	9-13 ∼I
15.	$j = k$	2,8,14 DS
16.	$(\mathcal{Q}(\vec{x}, j) \wedge (\forall z < j)\sim\mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, k) \wedge (\forall z < k)\sim\mathcal{Q}(\vec{x}, z)) \rightarrow j = k$	1-15 →I
17.	$\forall v[(\mathcal{Q}(\vec{x}, j) \wedge (\forall z < j)\sim\mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, v) \wedge (\forall z < v)\sim\mathcal{Q}(\vec{x}, z)) \rightarrow j = v]$	16 ∀I
18.	$\forall u\forall v[(\mathcal{Q}(\vec{x}, u) \wedge (\forall z < u)\sim\mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, v) \wedge (\forall z < v)\sim\mathcal{Q}(\vec{x}, z)) \rightarrow u = v]$	17 ∀I

So under the condition $\exists v\mathcal{Q}(\vec{x}, v)$, we have $\exists! v[\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v)\sim\mathcal{Q}(\vec{x}, v)]$. Thus we may define functions for minimization and bounded minimization under revised conditions. Let,

Def$[\mu v\mathcal{Q}(\vec{x}, v)]$ PA $\vdash v = \mu v\mathcal{Q}(\vec{x}, v) \leftrightarrow [\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v)\sim\mathcal{Q}(\vec{x}, v)]$

(i) PA $\vdash \exists v[\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v)\sim\mathcal{Q}(\vec{x}, v)]$.

(ii) $\forall u\forall v[(\mathcal{Q}(\vec{x}, u) \wedge (\forall z < u)\sim\mathcal{Q}(\vec{x}, z) \wedge \mathcal{Q}(\vec{x}, v) \wedge (\forall z < v)\sim\mathcal{Q}(\vec{x}, z)) \rightarrow u = v]$

But given T13.18, these conditions are met so long as PA $\vdash \exists v\mathcal{Q}(\vec{x}, v)$.

And,

Def$[(\mu y \leq z)\mathcal{Q}(\vec{x}, z, y)]$ PA $\vdash v = (\mu y \leq z)\mathcal{Q}(\vec{x}, z, y) \leftrightarrow v = \mu y[y = z \vee \mathcal{Q}(\vec{x}, z, y)]$

Let $m(\vec{x}, z) = \mu y[y = z \vee \mathcal{Q}(\vec{x}, z, y)]$ then we require,

(i) PA $\vdash \exists v(v = m(\vec{x}, z))$

(ii) PA $\vdash \forall u\forall v([u = m(\vec{x}, z) \wedge v = m(\vec{x}, z)] \rightarrow u = v)$

These conditions are trivially met so long as $m(\vec{x}, z)$ is defined; and for this, the existential condition, PA $\vdash \exists y[y = z \vee \mathcal{Q}(\vec{x}, z, y)]$ follows immediately from PA $\vdash z = z$; so the conditions for bounded minimization are automatically satisfied.

Given these notions, we may write down some immediate, simple results.

*T13.19. Let $m(\vec{x}) = \mu v \mathcal{Q}(\vec{x}, v)$; then,

(a) PA $\vdash \mathcal{Q}(\vec{x}, m(\vec{x})) \wedge (\forall z < m(\vec{x})) {\sim} \mathcal{Q}(\vec{x}, z)$

(b) PA $\vdash \mathcal{Q}(\vec{x}, m(\vec{x}))$

(c) PA $\vdash (\forall z < m(\vec{x})) {\sim} \mathcal{Q}(\vec{x}, z)$

(d) PA $\vdash \mathcal{Q}(\vec{x}, v) \rightarrow m(\vec{x}) \leq v$

Because it is always possible to switch bound variables so that $\mathcal{Q}$ is converted to an equivalent $\mathcal{Q}'$ whose bound variables do not overlap with variables free in $m(\vec{x})$, we simply assume $m(\vec{x})$ is free for v in $\mathcal{Q}(\vec{x}, v)$ (and we will generally make this move). Thus (a) follows from the definition $v = m(\vec{x}) \leftrightarrow [\mathcal{Q}(\vec{x}, v) \wedge (\forall z < v) {\sim} \mathcal{Q}(\vec{x}, v)]$ with v instantiated to $m(\vec{x})$ together with $m(\vec{x}) = m(\vec{x})$. Both conjuncts, and so (b) and (c) follow from (a). And (d) can be done in eight or nine lines with (c).

Of these, (a) - (c) simply observe that the definition applies to the function defined. From (d), the least v such that $\mathcal{Q}(\vec{x}, v)$ is always $\leq$ an arbitrary v such that $\mathcal{Q}(\vec{x}, v)$.

In addition, a couple of results for bounded minimization.

T13.20. The following result in PA,

(a) PA $\vdash (\mu y \leq \emptyset) \mathcal{Q}(\vec{x}, \emptyset, y) = \emptyset$

(b) If PA $\vdash (\exists v \leq t(u)) \mathcal{Q}(\vec{x}, u, v)$ then (i) PA defines $\mu v \mathcal{Q}(\vec{x}, u, v)$ and (ii) PA $\vdash (\mu v \leq t(u)) \mathcal{Q}(\vec{x}, u, v) = \mu v \mathcal{Q}(\vec{x}, u, v)$.

Hints: (a) follows easily from the definition. For (b), the existential for (i) follows simply from $(\exists v \leq t(u)) \mathcal{Q}(\vec{x}, u, v)$. For (ii),

1.	$(\exists v \le t(u))\mathcal{Q}(\vec{x},u,v)$	P
2.	$n(\vec{x},u) = (\mu v \le t(u))\mathcal{Q}(\vec{x},u,v)$	abv
3.	$n(\vec{x},u) = \mu v[v = t(u) \vee \mathcal{Q}(\vec{x},u,v)]$	2 def
4.	$n(\vec{x},u) = t(u) \vee \mathcal{Q}(\vec{x},u,n(\vec{x},u))$	3 T13.19b
5.	$\mathcal{Q}(\vec{x},u,a)$	A (g 1($\exists$E))
6.	$a \le t(u)$	
7.	$a < t(u) \vee a = t(u)$	6 T13.13l
8.	$a = t(u)$	A (g 7$\vee$E)
9.	$t(u) = n(\vec{x},u) \vee t(u) \neq n(\vec{x},u)$	T3.1
10.	$t(u) = n(\vec{x},u)$	A (g 9$\vee$E)
11.	$\mathcal{Q}(\vec{x},u,t(u))$	5,8 =E
12.	$\mathcal{Q}(\vec{x},u,n(\vec{x},u))$	11,10 =E
13.	$t(u) \neq n(\vec{x},u)$	A (g 9$\vee$E)
14.	$\mathcal{Q}(\vec{x},u,n(\vec{x},u))$	4,13 DS
15.	$\mathcal{Q}(\vec{x},u,n(\vec{x},u))$	9,10-12,13-14 $\vee$E
16.	$a < t(u)$	A (g 7$\vee$E)
17.	$a = t(u) \vee \mathcal{Q}(\vec{x},u,a)$	5 $\vee$I
18.	$n(\vec{x},u) \le a$	17 T13.19d
19.	$n(\vec{x},u) < t(u)$	18,16 T13.13c
20.	$n(\vec{x},u) \neq t(u)$	19 T13.13r
21.	$\mathcal{Q}(\vec{x},u,n(\vec{x},u))$	4,20 DS
22.	$\mathcal{Q}(\vec{x},u,n(\vec{x},u))$	7,8-15,16-21 $\vee$E
23.	$(\forall w < n(\vec{x},u))\sim[w = t(u) \vee \mathcal{Q}(\vec{x},u,w)]$	3 T13.19c
24.	$l < n(\vec{x},u)$	A (g ($\forall$I))
25.	$\sim[l = t(u) \vee \mathcal{Q}(\vec{x},u,l)]$	23,24 ($\forall$E)
26.	$l \neq t(u) \wedge \sim\mathcal{Q}(\vec{x},u,l)$	25 DeM
27.	$\sim\mathcal{Q}(\vec{x},u,l)$	26 $\wedge$E
28.	$(\forall w < n(\vec{x},u))\sim\mathcal{Q}(\vec{x},u,w)$	24-27 ($\forall$I)
29.	$\mathcal{Q}(\vec{x},u,n(\vec{x},u)) \wedge (\forall w < n(\vec{x},u))\sim\mathcal{Q}(\vec{x},u,w)$	22,28 $\wedge$I
30.	$n(\vec{x},u) = \mu v\mathcal{Q}(\vec{x},u,v)$	29 def
31.	$n(\vec{x},u) = \mu v\mathcal{Q}(\vec{x},u,v)$	1,5-30 ($\exists$E)
32.	$(\mu v \le t(u))\mathcal{Q}(\vec{x},u,v) = \mu v\mathcal{Q}(\vec{x},u,v)$	31 abv

So a is some object less than or equal to the bound $t(u)$ such that $\mathcal{Q}(\vec{x},u,a)$. It is clear enough that the least $\mathcal{Q}$ under the bound — $n(\vec{x},u)$ is same as $\mu v\mathcal{Q}(\vec{x},u,v)$ when $n(\vec{x},u)$ is other than $t(u)$. The most interesting case is the one at (10) where $n(\vec{x},u)$ is equal to the bound $t(u)$; then it remains that $\mathcal{Q}(\vec{x},u,n(\vec{x},u))$ because

$a = t(u)$ and $\mathcal{Q}(\vec{x}, u, a)$.

From, T13.20a it does not matter about $\mathcal{Q}$, the least y under the bound $\emptyset$ is always $\emptyset$. T13.20b converts between a bounded minimization and one without a bound; thus when T13.20b applies, results from from T13.19 for unbounded minimization apply to the bounded case.

*E13.7. Produce the quick derivation to show T13.19d.

E13.8. Complete the unfinished parts of T13.20.

13.3.2 Definitions for recursive functions

We now set out to show that PA defines relations and functions corresponding to recursive relations and functions. Insofar as we understand what a theorem of PA *is*, not all of the *demonstrations* are required to *understand* the argument — and some may obscure the overall flow. Thus, for our main argument, we often list results (with hints), shifting demonstrations into exercises and answers to exercises. To retain demonstration of results, a great many exercises are in fact worked in the answers section. Since the only constant in $\mathcal{L}_{\text{NT}}$ is $\emptyset$, there is no need to reserve letters for constants. Thus it is convenient to suppose that all of $a \ldots z$ are variables of the language.

The core result

The main argument is an induction on the sequence of recursive functions. However, with an eye to the β-function, we begin showing that PA defines remainder $rm(m, n)$ and quotient $qt(m, n)$ functions corresponding to $\mathsf{m}/(\mathsf{n}+1)$. Division is by $\mathsf{n}+1$ to avoid the possibility of division by zero.[10]

**Def*[*rm*] Let PA $\vdash v = rm(m, n) \leftrightarrow (\exists w \leq m)[m = Sn \times w + v \wedge v < Sn]$.

(i) PA $\vdash \exists x(\exists w \leq m)[m = Sn \times w + x \wedge x < Sn]$. Hint: This is an argument by IN on m. It is easy to show $\exists x(\exists w \leq \emptyset)[\emptyset = Sn \times w + x \wedge x < Sn]$, from $\emptyset = Sn \times \emptyset + \emptyset \wedge \emptyset < Sn$ with (∃I) and ∃I. Then, for the main argument, for the remainder k, $k < n \vee k = n$. In the first case Sj is divided by leaving

[10]A choice is made: Another option is define the functions with an arbitrary value for division by zero. Our selection makes for somewhat unintuitive statements of that which is intuitively true — rather than (relatively) intuitive statements including that which is intuitively undefined or false.

the quotient l the same, and incrementing k; in the second case Sj is divided by Sl with remainder zero.

(ii) PA $\vdash \forall x \forall y[((\exists w \leq m)[m = Sn \times w + x \wedge x < Sn] \wedge (\exists w \leq m)[m = Sn \times w + y \wedge y < Sn]) \rightarrow x = y]$. Hint: This does not require IN, but is an involved derivation all the same. Once you instantiate the bounded existential quantifiers to quotients p with remainder j and q with remainder k, you have $p < q \vee p = q \vee q < p$. When $p = q$, $j = k$ follows easily with cancellation for addition. And the other cases contradict. So, if $p < q$, you will be able to set up an l such that $Sl + p = q$, and show $j \not< Sn$. And similarly in the other case.

Def$[qt]$ Let PA $\vdash v = qt(m,n) \leftrightarrow m = Sn \times v + rm(m,n)$.

(i) PA $\vdash \exists x[m = Sn \times x + rm(m,n)]$. Hint: By =I, $rm(m,n) = rm(m,n)$; so with *Def*$[rm]$, $(\exists w \leq m)[m = Sn \times w + rm(m,n) \wedge rm(m,n) < Sn]$; and the result follows easily.

(ii) PA $\vdash \forall x \forall y[(m = Sn \times x + rm(m,n) \wedge m = Sn \times y + rm(m,n)) \rightarrow x = y]$. Hint: This is easy with cancellation laws for addition and multiplication.

Def$[\beta]$ PA $\vdash \beta(p,q,i) = rm(p, q \times Si)$.

Since this is a composition of functions, immediate from T13.17.

Observe that, from the definition, PA $\vdash v = \beta(p,q,i) \leftrightarrow (\exists w \leq p)[p = S(q \times Si) \times w + v \wedge v < S(q \times Si)]$, which is to say PA $\vdash v = \beta(p,q,i) \leftrightarrow \mathscr{B}(p,q,i,v)$, where $\mathscr{B}$ is the original formula to express the beta function.

And now our main argument that PA defines relations and functions corresponding to recursive relations and functions. The main result is for functions; relations follow as an easy corollary. But we shall not be able to show that PA defines relations and functions corresponding to *all* the recursive relations and functions: Say an application of regular minimization to generate $\mathsf{f}(\vec{\mathsf{x}})$ from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$ is (PA) *friendly* just in case PA $\vdash \exists y \mathscr{G}(\vec{x}, y, \emptyset)$ where $\mathscr{G}(\vec{x}, y, v)$ is the original formula that expresses and captures $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$; and a recursive function is (PA) *friendly* just in case it is an initial function or arises by applications of composition, recursion or friendly regular minimization. Observe that all *primitive* recursive functions are automatically friendly insofar as they involve no applications of minimization at all.

*T13.21. For any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and original formula $\mathcal{R}(\vec{x}, v)$ by which it is expressed and captured, PA defines a function $\mathbb{r}(\vec{x})$ such that PA $\vdash v = \mathbb{r}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, v)$.

By induction on the sequence of recursive functions.

Basis: $\mathsf{r}_0(\vec{\mathsf{x}})$ is an initial function $\mathsf{suc}(\mathsf{x})$, $\mathsf{zero}(\mathsf{x})$ or $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \ldots \mathsf{x}_{\mathsf{j}})$.

(s) $\mathsf{r}_0(\vec{\mathsf{x}})$ is $\mathsf{suc}(\mathsf{x})$. Let PA $\vdash v = \mathbb{suc}(x) \leftrightarrow Sx = v$. But $Sx = v$ is the original formula $Suc(x, v)$ by which $\mathsf{suc}(\mathsf{x})$ is expressed and captured; so PA $\vdash v = \mathbb{suc}(x) \leftrightarrow Suc(x, v)$. And by reasoning as follows,

1.	$Sx = Sx$	=I
2.	$\exists y(Sx = y)$	1 ∃I

1.	$Sx = j \wedge Sx = k$	A ($g \to$I)
2.	$Sx = j$	1 ∧E
3.	$Sx = k$	1 ∧E
4.	$j = k$	2,3 =E
5.	$(Sx = j \wedge Sx = k) \to j = k$	1-4 →I
6.	$\forall z[(Sx = j \wedge Sx = z) \to j = z]$	5 ∀I
7.	$\forall y \forall z[(Sx = y \wedge Sx = z) \to y = z]$	6 ∀I

PA $\vdash \exists! y(Sx = y)$. So PA defines $\mathbb{suc}(x)$.

(z) $\mathsf{r}_0(\vec{\mathsf{x}})$ is $\mathsf{zero}(\mathsf{x})$. Let PA $\vdash v = \mathbb{zero}(x) \leftrightarrow x = x \wedge v = \emptyset$. Then PA $\vdash v = \mathbb{zero}(x) \leftrightarrow Zero(x, v)$. And by (homework) PA defines $\mathbb{zero}(x)$.

(i) $\mathsf{r}_0(\vec{\mathsf{x}})$ is $\mathsf{idnt}^{\mathsf{j}}_{\mathsf{k}}(\mathsf{x}_1 \ldots \mathsf{x}_{\mathsf{j}})$. Let PA $\vdash v = \mathbb{idnt}^j_k(x_1 \ldots x_j) \leftrightarrow (x_1 = x_1 \wedge \ldots \wedge x_j = x_j) \wedge x_k = v$. Then PA $\vdash v = \mathbb{idnt}^j_k(x_1 \ldots x_j) \leftrightarrow Idnt^j_k(x_1 \ldots x_j, v)$. And by (homework) PA defines $\mathbb{idnt}^j_k(x_1 \ldots x_j)$.

Assp: For any i, $0 \leq i < k$, and $\mathsf{r}_{\mathsf{i}}(\vec{\mathsf{x}})$ with $\mathcal{R}_i(\vec{x}, v)$, PA defines $\mathbb{r}_i(\vec{x})$ such that PA $\vdash v = \mathbb{r}_i(\vec{x}) \leftrightarrow \mathcal{R}_i(\vec{x}, v)$.

Show: PA defines $\mathbb{r}_k(\vec{x})$ such that PA $\vdash v = \mathbb{r}_k(\vec{x}) \leftrightarrow \mathcal{R}_k(\vec{x}, v)$.

$\mathsf{r}_{\mathsf{k}}(\vec{\mathsf{x}})$ is either an initial function or arises by composition, recursion or PA friendly regular minimization. If $\mathsf{r}_{\mathsf{k}}(\vec{\mathsf{x}})$ is an initial function, then reason as in the basis. So suppose one of the other cases.

(c) $\mathsf{r}_{\mathsf{k}}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ is $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{g}(\vec{\mathsf{y}}), \vec{\mathsf{z}})$ for some $\mathsf{h}_{\mathsf{i}}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ and $\mathsf{g}_{\mathsf{j}}(\vec{\mathsf{y}})$ where $\mathsf{i}, \mathsf{j} < \mathsf{k}$. By assumption PA defines $\mathbb{h}(\vec{x}, w, \vec{z})$ such that PA $\vdash v = \mathbb{h}(\vec{x}, w, \vec{z}) \leftrightarrow \mathcal{H}(\vec{x}, w, \vec{z}, v)$ and PA defines $\mathbb{g}(\vec{y})$ such that PA $\vdash w = \mathbb{g}(\vec{y}) \leftrightarrow \mathcal{G}(\vec{y}, w)$. Let PA $\vdash \mathbb{r}_k(\vec{x}, \vec{y}, \vec{z}) = \mathbb{h}(\vec{x}, \mathbb{g}(\vec{y}), \vec{z})$. Then by T13.17 PA defines $\mathbb{r}_k$. And, where the original $\mathcal{R}_k$ is of the sort, $\exists w[\mathcal{G}(\vec{y}, w) \wedge \mathcal{H}(\vec{x}, w, \vec{z}, v)]$, PA $\vdash v = \mathbb{r}_k(\vec{x}, \vec{y}, \vec{z}) \leftrightarrow \mathcal{R}_k(\vec{x}, \vec{y}, \vec{z}, v)$. Thus, dropping $\vec{x}$ and $\vec{z}$ and reducing $\vec{y}$ to a single variable,

1.	$\mathbb{r}(y) = \mathbb{h}(\mathbb{g}(y))$	def
2.	$v = \mathbb{h}(w) \leftrightarrow \mathcal{H}(w, v)$	by assp
3.	$w = \mathbb{g}(y) \leftrightarrow \mathcal{G}(y, w)$	by assp
4.	$v = \mathbb{r}(y)$	A ($g \leftrightarrow$I)
5.	$v = \mathbb{h}(\mathbb{g}(y))$	1,4 =E
6.	$\mathbb{g}(y) = \mathbb{g}(y)$	=I
7.	$\mathbb{g}(y) = \mathbb{g}(y) \leftrightarrow \mathcal{G}(y, \mathbb{g}(y))$	3 ∀E
8.	$\mathcal{G}(y, \mathbb{g}(y))$	7,6 ↔E
9.	$\mathbb{h}(\mathbb{g}(y)) = \mathbb{h}(\mathbb{g}(y))$	=I
10.	$\mathbb{h}(\mathbb{g}(y)) = \mathbb{h}(\mathbb{g}(y)) \leftrightarrow \mathcal{H}(\mathbb{g}(y), \mathbb{h}(\mathbb{g}(y)))$	2 ∀E
11.	$\mathcal{H}(\mathbb{g}(y), \mathbb{h}(\mathbb{g}(y)))$	10,9 ↔E
12.	$\mathcal{H}(\mathbb{g}(y), v)$	11,5 =E
13.	$\mathcal{G}(y, \mathbb{g}(y)) \wedge \mathcal{H}(\mathbb{g}(y), v)$	8,12 ∧I
14.	$\exists w[\mathcal{G}(y, w) \wedge \mathcal{H}(w, v)]$	13 ∃I
15.	$\exists w[\mathcal{G}(y, w) \wedge \mathcal{H}(w, v)]$	A ($g \leftrightarrow$I)
16.	$\mathcal{G}(y, j) \wedge \mathcal{H}(j, v)$	A (g 15∃E)
17.	$j = \mathbb{g}(y) \leftrightarrow \mathcal{G}(y, j)$	3 ∀E
18.	$\mathcal{G}(y, j)$	16 ∧E
19.	$j = \mathbb{g}(y)$	17,18 ↔E
20.	$v = \mathbb{h}(j) \leftrightarrow \mathcal{H}(j, v)$	2 ∀E
21.	$\mathcal{H}(j, v)$	16 ∧E
22.	$v = \mathbb{h}(j)$	20,21 ↔E
23.	$v = \mathbb{h}(\mathbb{g}(y))$	22,19 =E
24.	$v = \mathbb{r}(y)$	1,23 =E
25.	$v = \mathbb{r}(y)$	15,16-24 ∃E
26.	$v = \mathbb{r}(y) \leftrightarrow \exists w[\mathcal{G}(y, w) \wedge \mathcal{H}(w, v)]$	4-14,15-25 ↔I

In the first subderivation, as usual, we suppose that quantifiers are arranged so that substitutions are allowed — and in particular so that $\mathbb{g}(y)$ is free for w in $\mathcal{H}(w, v)$ and $\mathcal{G}(y, w)$. Thus, with dropped variables restored we have, PA $\vdash v = \mathbb{r}_k(\vec{x}, \vec{y}, \vec{z}) \leftrightarrow \exists w[\mathcal{G}(\vec{y}, w) \wedge \mathcal{H}(\vec{x}, w, \vec{z}, v)]$ which is to say, PA $\vdash v = \mathbb{r}_k(\vec{x}) \leftrightarrow \mathcal{R}_k(\vec{x}, v)$.

(r) $\mathsf{r}_k(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from some $\mathsf{g}_i(\vec{\mathsf{x}})$ and $\mathsf{h}_j(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ where $\mathsf{i}, \mathsf{j} < \mathsf{k}$. By assumption PA defines $\mathbb{g}(\vec{x})$ such that PA $\vdash v = \mathbb{g}(\vec{x}) \leftrightarrow \mathcal{G}(\vec{x}, v)$ and PA defines $\mathbb{h}(\vec{x}, y, u)$ such that PA $\vdash v = \mathbb{h}(\vec{x}, y, u) \leftrightarrow \mathcal{H}(\vec{x}, y, u, v)$. Let PA $\vdash z = \mathbb{r}_k(\vec{x}, y) \leftrightarrow$

$$\exists p \exists q[\beta(p, q, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < y)\mathbb{h}(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \wedge \beta(p, q, y) = z]$$

By the argument of the next section, PA defines $\mathbb{r}(\vec{x}, y)$. And where the original $\mathcal{R}(\vec{x}, y, z) =$

$\exists p \exists q\{\exists v[\mathcal{B}(p,q,\emptyset,v) \wedge \mathcal{G}(\vec{x},v)] \wedge (\forall i < y)\exists u \exists v[\mathcal{B}(p,q,i,u) \wedge \mathcal{B}(p,q,Si,v) \wedge \mathcal{H}(\vec{x},i,u,v)] \wedge \mathcal{B}(p,q,y,z)\}$

we require PA $\vdash z = \mathbb{r}_k(\vec{x}, y) \leftrightarrow \mathcal{R}_k(\vec{x}, y, z)$. Here is the argument from left to right.

1.	$v = \beta(p,q,i) \leftrightarrow \mathcal{B}(p,q,i,v)$	def β
2.	$v = \mathscr{g}(\vec{x}) \leftrightarrow \mathcal{G}(\vec{x}, v)$	assp
3.	$v = \mathbb{h}(\vec{x}, y, u) \leftrightarrow \mathcal{H}(\vec{x}, y, u, v)$	assp
4.	$z = \mathbb{r}(\vec{x}, y)$	A ($g \rightarrow$I)
5.	$\exists p \exists q[\beta(p,q,\emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathbb{h}(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = z]$	4 def $\mathbb{r}$
6.	$\beta(a,b,\emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathbb{h}(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si) \wedge \beta(a,b,y) = z$	A (g 5$\exists$E)
7.	$\beta(a,b,\emptyset) = \mathscr{g}(\vec{x})$	6 $\wedge$E
8.	$\mathcal{G}(\vec{x}, \mathscr{g}(\vec{x}))$	from 2
9.	$\mathcal{B}(a,b,\emptyset,\beta(a,b,\emptyset))$	from 1
10.	$\mathcal{B}(a,b,\emptyset,\mathscr{g}(\vec{x}))$	7,9 =E
11.	$\mathcal{B}(a,b,\emptyset,\mathscr{g}(\vec{x})) \wedge \mathcal{G}(\vec{x}, \mathscr{g}(\vec{x}))$	10,8 $\wedge$I
12.	$\exists v[\mathcal{B}(a,b,\emptyset,v) \wedge \mathcal{G}(\vec{x}, v)]$	11 $\exists$I
13.	$(\forall i < y)\mathbb{h}(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si)$	6 $\wedge$E
14.	$l < y$	A (g ($\forall$I))
15.	$\mathbb{h}(\vec{x}, l, \beta(a,b,l)) = \beta(a,b,Sl)$	13,14 ($\forall$E)
16.	$\mathcal{B}(a,b,l,\beta(a,b,l))$	from 1
17.	$\mathcal{B}(a,b,Sl,\beta(a,b,Sl))$	from 1
18.	$\mathcal{H}(\vec{x}, l, \beta(a,b,l), \mathbb{h}(\vec{x}, l, \beta(a,b,l)))$	from 3
19.	$\mathcal{H}(\vec{x}, l, \beta(a,b,l), \beta(a,b,Sl))$	18,15 =E
20.	$\mathcal{B}(a,b,l,\beta(a,b,l)) \wedge \mathcal{B}(a,b,Sl,\beta(a,b,Sl)) \wedge \mathcal{H}(\vec{x}, l, \beta(a,b,l), \beta(a,b,Sl))$	16,17,19 $\wedge$I
21.	$\exists u \exists v[\mathcal{B}(a,b,l,u) \wedge \mathcal{B}(a,b,Sl,v) \wedge \mathcal{H}(\vec{x}, l, u, v)]$	20 $\exists$I
22.	$(\forall i < y)\exists u \exists v[\mathcal{B}(a,b,i,u) \wedge \mathcal{B}(a,b,Si,v) \wedge \mathcal{H}(\vec{x}, i, u, v)]$	14-21 ($\forall$I)
22.	$\beta(a,b,y) = z$	6 $\wedge$E
23.	$\mathcal{B}(a,b,y,\beta(a,b,y))$	from 1
24.	$\mathcal{B}(a,b,y,z)$	23,22 =E
25.	$\exists v[\mathcal{B}(a,b,\emptyset,v) \wedge \mathcal{G}(\vec{x}, v)] \wedge$ $(\forall i < y)\exists u \exists v[\mathcal{B}(a,b,i,u) \wedge \mathcal{B}(a,b,Si,v) \wedge \mathcal{H}(\vec{x}, i, u, v)] \wedge \mathcal{B}(a,b,y,z)$	12,22,24 $\wedge$I
26.	$\exists p \exists q\{\exists v[\mathcal{B}(p,q,\emptyset,v) \wedge \mathcal{G}(\vec{x}, v)] \wedge$ $(\forall i < y)\exists u \exists v[\mathcal{B}(p,q,i,u) \wedge \mathcal{B}(p,q,Si,v) \wedge \mathcal{H}(\vec{x}, i, u, v)] \wedge \mathcal{B}(p,q,y,z)\}$	25 $\exists$I
27.	$\mathcal{R}(\vec{x}, y, z)$	26 def
28.	$\mathcal{R}(\vec{x}, y, z)$	5,6-27 $\exists$E
29.	$z = \mathbb{r}(\vec{x}, y) \rightarrow \mathcal{R}(\vec{x}, y, z)$	4-28 $\rightarrow$I

The other direction is left as an exercise.

(m) $\mathsf{f}_k(\vec{\mathsf{x}})$ arises by friendly regular minimization from $\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})$. By assumption PA defines $\mathscr{g}(\vec{x}, y)$ such that PA $\vdash v = \mathscr{g}(\vec{x}, y) \leftrightarrow \mathcal{G}(\vec{x}, y, v)$ where $\mathcal{G}$ is the original formula to express and capture g. Let PA $\vdash \mathbb{r}_k(\vec{x}) = \mu y \mathcal{G}(\vec{x}, y, \emptyset)$. Since the minimization is friendly, PA $\vdash \exists y \mathcal{G}(\vec{x}, y, \emptyset)$; so by T13.19, PA defines $\mathbb{r}_k(\vec{x})$. And by definition, PA $\vdash v = \mathbb{r}_k(\vec{x}) \leftrightarrow \mathcal{G}(\vec{x}, v, \emptyset) \wedge (\forall y < v){\sim}\mathcal{G}(\vec{x}, y, \emptyset)$. So PA $\vdash$

$v = \mathscr{r}_k(\vec{x}) \leftrightarrow \mathcal{R}_k(\vec{x}, v)$.

Indct: For any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and the original formula $\mathcal{R}(\vec{x}, v)$ by which it is expressed and captured, PA defines a function $\mathscr{r}(\vec{x})$ such that PA $\vdash v = \mathscr{r}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, v)$ (subject to the recursion clause) .

*E13.9. Complete the justifications for *Def*[*rm*] and *Def*[*qt*].

*E13.10. Complete the unfinished cases to T13.21. You should set up the entire induction, but may refer to the text as the text refers unfinished cases to homework.

The Recursion Clause

We turn now to a series of results with the aim of showing that PA defines $\mathscr{r}$ in the case when r arises by recursion. This will require a series of definitions and results in PA. Some of the functions so defined parallel ones that will result from recursive functions. However, insofar as we have not yet proved the core result, we cannot use it! So we are showing directly that PA gives the required results.

Uniqueness. It will be easiest to begin with the uniqueness clause. Where $\mathcal{F}(\vec{x}, y, v)$ is our formula,

$$\exists p \exists q[\beta(p,q,\emptyset) = \mathscr{g}(\vec{x}) \wedge (\forall i < y)\mathscr{h}(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = z]$$

we want PA $\vdash \forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \wedge \mathcal{F}(\vec{x}, y, n)) \to m = n]$. The argument is structured very much as for the parallel uniqueness case in Q (T12.12) except that the argument is in PA and so by IN, and uniqueness conditions are simplified by the use of function symbols. The argument is simplified — but that does not mean that it is simple!

T13.22. With $\mathcal{F}(\vec{x}, y, v)$ as described above, PA $\vdash \forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \wedge \mathcal{F}(\vec{x}, y, n)) \to m = n]$.

For the zero case you need to show $\forall m \forall n[(\mathcal{F}(\vec{x}, \emptyset, m) \wedge \mathcal{F}(\vec{x}, \emptyset, n)) \to m = n]$. This is simple enough and left as an exercise. Given the zero case, here is the main argument by IN.

First theorems of chapter 13

T13.1 For any recursively axiomatized theory T whose language includes $\mathscr{L}_{NT}$, $\mathscr{G}$ is true iff it is unprovable in T (iff $T \nvdash \mathscr{G}$).

T13.2 If T is a recursively axiomatized sound theory whose language includes $\mathscr{L}_{NT}$, then T is negation incomplete.

T13.3 Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathscr{G} \leftrightarrow \sim\exists x Prft(x, \overline{\ulcorner\mathscr{G}\urcorner})$.

T13.4 If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathscr{G}$.

T13.5 If T is an ω-consistent, recursively axiomatized theory extending Q, then $T \nvdash \sim\mathscr{G}$.

T13.6 Let T be any recursively axiomatized theory extending Q; then $T \vdash \mathscr{R} \leftrightarrow \sim\exists x RPrf(x, \overline{\ulcorner\mathscr{R}\urcorner})$.

T13.7 If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \mathscr{R}$.

T13.8 If T is a consistent, recursively axiomatized theory extending Q, then $T \nvdash \sim\mathscr{R}$.

T13.9 Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic of provability, $T \vdash Cont \to \sim Prvt(\overline{\ulcorner\mathscr{G}\urcorner})$.

T13.10 Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions, if T is consistent, $T \nvdash Cont$.

T13.11 Let T be a recursively axiomatized theory extending Q. Then supposing T satisfies the derivability conditions and so the K4 logic of provability, $T \vdash Cont \leftrightarrow \sim Prvt(\overline{\ulcorner Cont\urcorner})$.

T13.12 Suppose T is a recursively axiomatized theory extending Q. Then if $T \vdash \mathscr{P}$, then $T \vdash \Box\mathscr{P}$.

T13.13 This lists a number of straightforward theorems of PA.

T13.14 For a defined relation symbol, function symbol or constant, with its associated axiom and conversion procedure, $T' \vdash \mathscr{F}' \leftrightarrow \mathscr{F}$.

T13.15 For a defined relation symbol, function symbol or constant, with its associated axiom, and any formula $\mathscr{F}$ in the original language, $T' \vdash \mathscr{F}$ iff $T \vdash \mathscr{F}$.

T13.16 For some defined relation symbols, function symbols or constants, with their associated axioms and conversion procedures, $T' \vdash \mathscr{F}'$ iff $T \vdash \mathscr{F}$.

T13.17 If PA defines some $h(\vec{x}, w, \vec{z})$ and $g(\vec{y})$, then PA defines $f(\vec{x}, \vec{y}, \vec{z}) = h(\vec{x}, g(\vec{y}), \vec{z})$.

T13.18 If PA $\vdash \exists v\mathscr{Q}(\vec{x}, v)$, then PA $\vdash \exists! v[\mathscr{Q}(\vec{x}, v) \wedge (\forall z < v)\sim\mathscr{Q}(\vec{x}, v)]$.

T13.19 Where $m(\vec{x}) = \mu v\mathscr{Q}(\vec{x}, v)$, (a) PA $\vdash \mathscr{Q}(\vec{x}, m(\vec{x})) \wedge (\forall z < m(\vec{x}))\sim\mathscr{Q}(\vec{x}, z)$; (b) PA $\vdash \mathscr{Q}(\vec{x}, m(\vec{x}))$; (c) PA $\vdash (\forall z < m(\vec{x}))\sim\mathscr{Q}(\vec{x}, z)$; (d) PA $\vdash \mathscr{Q}(\vec{x}, v) \to m(\vec{x}) \leq v$.

T13.20 (a) PA $\vdash (\mu y \leq \emptyset)\mathscr{Q}(\vec{x}, \emptyset, y) = \emptyset$; (b) if PA $\vdash (\exists v \leq \mathscr{t}(u))\mathscr{Q}(\vec{x}, u, v)$ then (i) PA defines $\mu v\mathscr{Q}(\vec{x}, u, v)$ and (ii) PA $\vdash (\mu v \leq \mathscr{t}(u))\mathscr{Q}(\vec{x}, u, v) = \mu v\mathscr{Q}(\vec{x}, u, v)$.

T13.21 For any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and original formula $\mathscr{R}(\vec{x}, v)$ by which it is expressed and captured, PA defines a function $\mathscr{r}(\vec{x})$ such that PA $\vdash v = \mathscr{r}(\vec{x}) \leftrightarrow \mathscr{R}(\vec{x}, v)$. This theorem depends on conditions for the recursion clause and so on T13.22 and T13.31.

1.	$\forall m \forall n[(\mathscr{F}(\vec{x}, \emptyset, m) \wedge \mathscr{F}(\vec{x}, \emptyset, n)) \rightarrow m = n]$	zero case
2.	$\forall m \forall n[(\mathscr{F}(\vec{x}, j, m) \wedge \mathscr{F}(\vec{x}, j, n)) \rightarrow m = n]$	A ($g \rightarrow$I)
3.	$\mathscr{F}(\vec{x}, Sj, u) \wedge \mathscr{F}(\vec{x}, Sj, v)$	A ($g \rightarrow$I)
4.	$\exists p \exists q[\beta(p, q, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \wedge \beta(p, q, Sj) = u]$	3 $\wedge$E
5.	$\exists p \exists q[\beta(p, q, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \wedge \beta(p, q, Sj) = v]$	3 $\wedge$E
6.	$\beta(a, b, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si) \wedge \beta(a, b, Sj) = u$	A (g 4$\exists$E)
7.	$\beta(a, b, \emptyset) = \mathbb{g}(\vec{x})$	6 $\wedge$E
8.	$(\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si)$	6 $\wedge$E
9.	$\beta(a, b, Sj) = u$	6 $\wedge$E
10.	$\beta(c, d, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(c, d, i)) = \beta(c, d, Si) \wedge \beta(c, d, Sj) = v$	A (g 5$\exists$E)
11.	$\beta(c, d, \emptyset) = \mathbb{g}(\vec{x})$	10 $\wedge$E
12.	$(\forall i < Sj)\mathbb{h}(\vec{x}, i, \beta(c, d, i)) = \beta(c, d, Si)$	10 $\wedge$E
13.	$\beta(c, d, Sj) = v$	10 $\wedge$E
14.	$j < Sj$	T13.13g
15.	$\mathbb{h}(\vec{x}, j, \beta(a, b, j)) = \beta(a, b, Sj)$	8,14 ($\forall$E)
16.	$\mathbb{h}(\vec{x}, j, \beta(c, d, j)) = \beta(c, d, Sj)$	12,14 ($\forall$E)
17.	$k < j$	A (g ($\forall$I))
18.	$k < Sj$	17, T13.13m
19.	$\mathbb{h}(\vec{x}, k, \beta(a, b, k)) = \beta(a, b, Sk)$	8,18 ($\forall$E)
20.	$(\forall i < j)\mathbb{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si)$	17-19 ($\forall$I)
21.	$\beta(a, b, j) = \beta(a, b, j)$	=I
22.	$\beta(a, b, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < j)\mathbb{h}(\vec{x}, i, \beta(a, b, i)) = \beta(a, b, Si) \wedge \beta(a, b, j) = \beta(a, b, j)$	7,20,21 $\wedge$I
23.	$\exists p \exists q[\beta(p, q, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < j)\mathbb{h}(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \wedge \beta(p, q, j) = \beta(a, b, j)]$	22 $\exists$I
24.	$\mathscr{F}(\vec{x}, j, \beta(a, b, j))$	23 abv
25.	$k < j$	A (g ($\forall$I))
26.	$k < Sj$	25, T13.13m
27.	$\mathbb{h}(\vec{x}, k, \beta(c, d, k)) = \beta(c, d, Sk)$	12,26 ($\forall$E)
28.	$(\forall i < j)\mathbb{h}(\vec{x}, i, \beta(c, d, i)) = \beta(c, d, Si)$	25-27 ($\forall$I)
29.	$\beta(c, d, j) = \beta(c, d, j)$	=I
30.	$\beta(c, d, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < j)\mathbb{h}(\vec{x}, i, \beta(c, d, i)) = \beta(c, d, Si) \wedge \beta(c, d, j) = \beta(c, d, j)$	11,28,29 $\wedge$I
31.	$\exists p \exists q[\beta(p, q, \emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < j)\mathbb{h}(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \wedge \beta(p, q, j) = \beta(c, d, j)]$	30 $\exists$I
32.	$\mathscr{F}(\vec{x}, j, \beta(c, d, j))$	31 abv
33.	$\beta(a, b, j) = \beta(c, d, j)$	2,24,32 $\forall$E
34.	$\mathbb{h}(\vec{x}, j, \beta(c, d, j)) = \beta(a, b, Sj)$	15,33 =E
35.	$\beta(a, b, Sj) = \beta(c, d, Sj)$	34,16 =E
36.	$u = v$	9,13,35 =E
37.	$u = v$	5,10-36 $\exists$E
38.	$u = v$	4,6-37 $\exists$E
39.	$(\mathscr{F}(\vec{x}, Sj, u) \wedge \mathscr{F}(\vec{x}, Sj, v)) \rightarrow u = v$	3-38 $\rightarrow$I
40.	$\forall m \forall n[(\mathscr{F}(\vec{x}, Sj, m) \wedge \mathscr{F}(\vec{x}, Sj, n)) \rightarrow m = n]$	39 $\forall$I
41.	$\forall m \forall n[(\mathscr{F}(\vec{x}, j, m) \wedge \mathscr{F}(\vec{x}, j, n)) \rightarrow m = n] \rightarrow \forall m \forall n[(\mathscr{F}(\vec{x}, Sj, m) \wedge \mathscr{F}(\vec{x}, Sj, n)) \rightarrow m = n]$	2-40 $\rightarrow$I
42.	$\forall y\{\forall m \forall n[(\mathscr{F}(\vec{x}, y, m) \wedge \mathscr{F}(\vec{x}, y, n)) \rightarrow m = n] \rightarrow \forall m \forall n[(\mathscr{F}(\vec{x}, Sy, m) \wedge \mathscr{F}(\vec{x}, Sy, n)) \rightarrow m = n]\}$	41 $\forall$I
43.	$\forall y \forall m \forall n[(\mathscr{F}(\vec{x}, y, m) \wedge \mathscr{F}(\vec{x}, y, n)) \rightarrow m = n]$	1,42 IN
44.	$\forall m \forall n[(\mathscr{F}(\vec{x}, y, m) \wedge \mathscr{F}(\vec{x}, y, n)) \rightarrow m = n]$	43 $\forall$E

As before, the key to this argument is attaining $\mathscr{F}(\vec{x}, j, \beta(a, b, j))$ and $\mathscr{F}(\vec{x}, j, \beta(c, d, j))$ on lines (24) and (32). From these the assumption on (2) comes into play, and the result follows with other equalities.

*E13.11. Complete the demonstration for T13.22 by completing the demonstration of the zero case.

Existence. Considerably more difficult is the existential condition. To show this, we must show the Chinese remainder theorem in PA. Again, we build by a series of results.

First, subtraction with cutoff. The definition is not recursive as before. However the effect is the same: $x \dot{-} y$ works like subtraction when $x \geq y$, and otherwise goes to $\emptyset$.

*$Def[\dot{-}]$ PA $\vdash v = x \dot{-} y \leftrightarrow x = y + v \vee (x < y \wedge v = \emptyset)$

(i) PA $\vdash \exists v[x = y + v \vee (x < y \wedge v = \emptyset)]$

(ii) PA $\vdash \forall m \forall n[([x = y+m \vee (x < y \wedge m = \emptyset)] \wedge [x = y+n \vee (x < y \wedge n = \emptyset)]) \rightarrow m = n]$

The proof of (i) and (ii) is left as an exercise. So PA defines ($\dot{-}$). And it proves a series of intuitive results.

*T13.23. The following result in PA:

(a) PA $\vdash a \geq b \rightarrow a = b + (a \dot{-} b)$

(b) PA $\vdash b \geq a \rightarrow a \dot{-} b = \emptyset$

(c) PA $\vdash a \dot{-} b \leq a$

*(d) PA $\vdash a > b \rightarrow a \dot{-} b > \emptyset$

(e) PA $\vdash a \dot{-} \emptyset = a$

(f) PA $\vdash Sa \dot{-} a = \overline{1}$

(g) PA $\vdash a > \emptyset \rightarrow a \dot{-} \overline{1} < a$

(h) PA $\vdash a \geq Sb \rightarrow a \dot{-} b = S(a \dot{-} Sb)$

(i) PA $\vdash a = Sa \dot{-} \overline{1}$

*(j) PA $\vdash a \geq c \rightarrow (a \mathbin{\dot{-}} c) + b = (a + b) \mathbin{\dot{-}} c$

(k) PA $\vdash (a \geq b \wedge b \geq c) \rightarrow a \mathbin{\dot{-}} (b \mathbin{\dot{-}} c) = (a \mathbin{\dot{-}} b) + c$

*(l) PA $\vdash (a \mathbin{\dot{-}} b) \mathbin{\dot{-}} c = a \mathbin{\dot{-}} (b + c)$

(m) PA $\vdash (a + c) \mathbin{\dot{-}} (b + c) = a \mathbin{\dot{-}} b$

*(n) PA $\vdash a \times (b \mathbin{\dot{-}} c) = a \times b \mathbin{\dot{-}} a \times c$

Hints. (d): with the assumption you can get both $a = Sj + b$ and $a = b + (a \mathbin{\dot{-}} b)$; then you have what you need with T6.66. (j): with the assumption $a \geq c$ you have also $a + b \geq c$; so that both $a = c + (a \mathbin{\dot{-}} c)$ and $a + b = c + [(a + b) \mathbin{\dot{-}} c]$; then =E and T6.66 do the work. (k): You can get this with a couple applications of (j). (l): First, $a \geq b + c \vee a < b + c$; in the second case, $a \geq b \vee a < b$; in each of these cases, both sides equal $\emptyset$; for the first main option, you will be able to show that $(b + c) + [(a \mathbin{\dot{-}} b) \mathbin{\dot{-}} c] = (b + c) + [a \mathbin{\dot{-}} (b + c)]$ and apply T6.66. (n): First $a = \emptyset \vee a > \emptyset$; in the first case, both sides equal $\emptyset$; then in the second case, $b \geq c \vee b < c$; again in the first of these cases, both sides equal $\emptyset$; in the last case, you will be able to show $ac + a(b \mathbin{\dot{-}} c) = ac + (ab \mathbin{\dot{-}} ac)$ and apply T6.66.

Many of these state standard results for subtraction — except where the inequalities are required to protect against cases when $a \mathbin{\dot{-}} b$ goes to $\emptyset$. (a) and (b) extract basic information from the definition upon which rest depend. (c) - (i) are simple subtraction facts. And (j) - (n) are some results for association and distribution.

Next *factor*. Again, consistent with remainder and quotient, we say $m|n$ when $m + 1$ divides n.

Def[|] PA $\vdash m|n \leftrightarrow \exists q(Sm \times q = n)$

Since factor is a relation, no condition is required over and above the axiom so that the definition is good as it stands. And, again, PA proves a series of results. These are reasonably intuitive. Observe, however that our choice to divide by $m + 1$ means that, as in T13.24a below, $\emptyset|a$.

*T13.24. The following result in PA:

(a) PA $\vdash \emptyset|a$

(b) PA $\vdash a|Sa$

(c) PA $\vdash a|\emptyset$

(d) PA $\vdash a|b \to a|(b \times c)$

(e) PA $\vdash (a > \emptyset \wedge b > \emptyset) \to [(a \dot{-} \overline{1})|c \wedge (b \dot{-} \overline{1})|d \to (ab \dot{-} \overline{1})|cd]$

(f) PA $\vdash (a|Sb \wedge b|c) \to a|c$

*(g) PA $\vdash a|b \to [a|(b + c) \leftrightarrow a|c]$

(h) PA $\vdash (b \geq c \wedge a|b) \to [a|(b \dot{-} c) \leftrightarrow a|c]$

(i) PA $\vdash b > a \to b \nmid Sa$

(j) PA $\vdash a|b \leftrightarrow rm(b, a) = \emptyset$

*(k) PA $\vdash rm[a + (y \times Sd), d] = rm(a, d)$

*(l) PA $\vdash Sd \times z \leq a \to z \leq qt(a, d)$

*(m) PA $\vdash a \geq y \times Sd \to rm[a \dot{-} (y \times Sd), d] = rm(a, d)$

Hints. (g): The assumption $a|b$ gives $Sa \times j = b$; then $a|(b + c)$ gives $Sa \times k = b + c$; you will have to show $j \leq k$ so that $l + j = k$; $a|c$ follows with these; then $a|c$ gives $Sa \times k = c$ and you will be able to substitute for both b and c to get $(Sa \times j) + (Sa \times k) = b + c$; the result follows with this. (k): From the assumption you have $a = (Sd \times j) + r \wedge r < Sd$; and if you assert $a + (y \times Sd) = a + (y \times Sd)$ by =I you should be able to show, $a + (y \times Sd) = Sd \times (j + y) + r \wedge r < Sd$; then with $j + y \leq a + (y \times Sd)$ you can apply (∃I) and the definition. (l): With $r = rm(a, d)$ and $q = qt(a, d)$ by *Def*[qt] you have $a = Sd \times q + r \wedge r < Sd$; assume $Sd \times z \leq a$ for →I and $z > q$ for ∼I; then you should be able to show $a < Sd \times z$ to contradict the assumption for →I. (m): Again let $r = rm(a, d)$ and $q = qt(a, d)$; then by *Def*[qt] you have $a = Sd \times q + r \wedge r < Sd$; assume $a \geq y \times Sd$ for →I; you should be able to show, $a \dot{-} (y \times Sd) = Sd(q \dot{-} y) + r \wedge r < Sd$ toward $(\exists w < a \dot{-} (y \times Sd))[a \dot{-} (y \times Sd) = Sd \times w + r \wedge r < Sd]$ by (∃I), to apply *Def*[rm].

So (a) (the successor of) $\emptyset$ divides any number; (b) (the successor of) a divides Sa; and (c) any number divides into $\emptyset$ zero times. (d) if a divides b then it divides $b \times c$; (e) where subtraction compensates for successor, if a divides c and b divides d, ab divides cd; and (f) if a divides Sb and (the successor of) b divides c, then a divides

c. (g) is like $(b + c)/a = b/a + c/a$ so that dividing the sum breaks into dividing the members; (h) is the comparable principle for subtraction. From (i) if $b > a$, then (the successor of) b does not divide Sa. (j) makes the obvious connection between reminder and factor. In (k) the remainder of the second part $(y \times Sd)$ is $\emptyset$ so that the remainder of the sum is just whatever there is from the first $rm(a, d)$; (m) is the comparable principle for subtraction. The intervening (l) is required for (m) and tells us that if z multiples of (the successor of) d come to $\leq a$, then $z \leq qt(a, d)$ — since the quotient maximizes the multiples of (the successor of) d that are $\leq a$.

And now PA defines relations *prime* and *relatively prime*. Prime has its usual sense. And numbers are relatively prime when they have no common divisor other than one — though they may not therefore individually be prime. Though division is by successor, these notions are given their usual sense by adjusting the numbers that are said to "divide."

Def[*Pr*] PA $\vdash Pr(n) \leftrightarrow \overline{1} < n \wedge \forall x[x|n \rightarrow (x = \emptyset \vee Sx = n)]$

Def[*Rp*] PA $\vdash Rp(a, b) \leftrightarrow \forall x[(x|a \wedge x|b) \rightarrow x = \emptyset]$

Since these are relations, no condition is required over and above the axioms. So $\overline{1}$ is relatively prime with anything, since the only number that divides both $\overline{1}$ and some b is (the successor of) $\emptyset$. Also, for any a, $a|\emptyset$ and $a|Sa$; so $\emptyset$ is relatively prime with $\overline{1}$; but it is not relatively prime with anything else, since when $a > \emptyset$, both $\emptyset$ and Sa are divided by a number other than (the successor of) $\emptyset$.

It will be helpful to introduce a couple of subsidiary notions. When $G(a, b, i)$ we say that i is *good*, and $d(a, b)$ is the *least* such good,

Def[G] PA $\vdash G(a, b, i) \leftrightarrow \exists x \exists y(ax + i = by)$

Def[d] PA $\vdash d(a, b) = \mu v[(a > \emptyset \wedge b > \emptyset) \rightarrow G(a, b, Sv)]$

(i) PA $\vdash \exists v[(a > \emptyset \wedge b > \emptyset) \rightarrow G(a, b, Sv)]$

Begin with $b = \emptyset \vee b > \emptyset$ and go for the existentially quantified goal. In the second case, there is some l such that $b = Sl$ and it is easy to show, $a \times \emptyset + b = b \times \overline{1}$ and generalize.

If a or b is not greater than $\emptyset$ then $d(a, b)$ is just $\emptyset$. Otherwise, the notion is more significant.

Again, PA proves a series of results. Observe again that if we are interested in whether a prime divides some b we are interested in whether $Pr(Sa) \wedge a|b$ since it is the successor that is divided into b.

*T13.25. The following result in PA:

(a) PA $\vdash \sim Pr(\emptyset)$

(b) PA $\vdash \sim Pr(\overline{1})$

(c) PA $\vdash Pr(\overline{2})$

*(d) PA $\vdash \forall x[x > \overline{1} \rightarrow \exists z(Pr(Sz) \wedge z|x)]$

*(e) PA $\vdash Rp(a,b) \leftrightarrow \sim\exists x[Pr(Sx) \wedge x|a \wedge x|b]$

(f) PA $\vdash \forall x \forall y[G(a,b,x) \rightarrow G(a,b,x \times y)]$

*(g) PA $\vdash (a > \emptyset \wedge b > \emptyset) \rightarrow \forall x \forall y[(G(a,b,x) \wedge G(a,b,y) \wedge x \geq y) \rightarrow G(a,b,x \dot{-} y)]$

*(h) PA $\vdash [Rp(a,b) \wedge a > \emptyset \wedge b > \emptyset] \rightarrow G(a,b,\overline{1})$

*(i) PA $\vdash [Pr(Sa) \wedge a|(b \times c)] \rightarrow (a|b \vee a|c)$

Hints. (c): This is straightforward with T13.24i. (d): You can do this by the second form of strong induction T13.13af; the zero case is trivial; to reach $\forall x\{(\forall y \leq x)[y > \overline{1} \rightarrow \exists z(Pr(Sz) \wedge z|y)] \rightarrow [Sx > \overline{1} \rightarrow \exists z(Pr(Sz) \wedge z|Sx)]\}$ assume $(\forall y \leq k)[y > \overline{1} \rightarrow \exists z(Pr(Sz) \wedge z|y)]$ and $Sk > \overline{1}$; then Sk is prime or not; if it is prime, the result is immediate; if it is not, you will be able to show $Sj \leq k$ and apply the assumption. (e): From left to right, under the assumption for $\leftrightarrow$I assume $\exists x[Pr(Sx) \wedge x|a \wedge x|b]$ and $Pr(Sj) \wedge j|a \wedge j|b$ for $\sim$I and $\exists$E; then you should be able to show that $\overline{1} < Sj$ and $\overline{1} \not< Sj$; in the other direction, under the assumption for $\leftrightarrow$I and then $j|a \wedge j|b$ for $\rightarrow$I, $j = \emptyset \vee j > \emptyset$ by T13.13f; the latter is impossible, which gives the result you want. (g): Under the assumptions $a > \emptyset \wedge b > \emptyset$ and then $G(a,b,i) \wedge G(a,b,j) \wedge i \geq j$ for $\rightarrow$I and then $ap + i = bq$ and $ar + j = bs$ for $\exists$E, starting with $(bq + bar) + (bsa \dot{-} bs) = (bq + bar) + (bsa \dot{-} bs)$ by =I, with some effort, you will be able to show $a[(p + bs) + (br \dot{-} r)] + (i \dot{-} j) = b[(q + ar) + (sa \dot{-} s)]$ and generalize. (i): Under the assumption $Pr(Sa) \wedge a|(b \times c)$ assume $a \nmid b$ with the idea of obtaining $a \nmid b \rightarrow a|c$ for Impl; set out to show $Rp(b, Sa)$ for an

application of T13.25h to get $\exists x \exists y[bx + \overline{1} = Sa \times y]$; with this, you will have $bp + \overline{1} = Sa \times q$ by $\exists$E; and you should be able to show $a|cbp$ and $a|(cbp + c)$ for an application of T13.24g.

T13.25h is important. But the argument is relatively complex; it has the following main stages.

1.	$[(a > \emptyset \wedge b > \emptyset) \rightarrow G(a,b,Sd(a,b))] \wedge (\forall y < d(a,b)) \sim [(a > \emptyset \wedge \wedge b > \emptyset) \rightarrow G(a,b,Sy)]$	def d
2.	$(a > \emptyset \wedge b > \emptyset) \rightarrow G(a,b,Sd(a,b))$	1 $\wedge$E
3.	$Rp(a,b) \wedge a > \emptyset \wedge b > \emptyset$	A ($g \rightarrow$I)
4.	$Rp(a,b)$	3 $\wedge$E
5.	$\forall x[(x\|a \wedge x\|b) \rightarrow x = \emptyset]$	4 def
6.	$a > \emptyset \wedge b > \emptyset$	4 $\wedge$E
7.	$G(a,b,Sd(a,b))$	2,6 $\rightarrow$E
8.	$G(a,b,a)$	[a]
9.	$G(a,b,b)$	[b]
10.	$\forall x[G(a,b,x) \rightarrow d(a,b)\|x]$	[c]
11.	$d(a,b)\|a$	8,10 $\forall$E
12.	$d(a,b)\|b$	9,10 $\forall$E
13.	$d(a,b)\|a \wedge d(a,b)\|b$	11,12 $\wedge$I
14.	$d(a,b) = \emptyset$	5,13 $\forall$E
15.	$G(a,b,\overline{1})$	7,14 =E
16.	$[Rp(a,b) \wedge a > \emptyset \wedge b > \emptyset] \rightarrow G(a,b,\overline{1})$	3-15 $\rightarrow$I

Hint. For (c) let $q = qt(i, d(a,b))$ and $r = rm(i, d(a,b))$ then from the definitions you have $i = (Sd(a,b) \times q) + r$ and $r < Sd(a,b)$ and from (1) of the main argument $(\forall y < d(a,b)) \sim [(a > \emptyset \wedge b > \emptyset) \rightarrow G(a,b,Sy)]$; then under the assumption $G(a,b,i)$ for $\rightarrow$I you should be able to show $G(a,b,i \dot{-} (Sd(a,b) \times q))$ using (6) from the main argument with (f) and (g); but also $i \dot{-} (Sd(a,b) \times q) = r$ so that $G(a,b,r)$. Now the assumption that r is a successor leads to contradiction; so $r = \emptyset$ and $d(a,b)|i$.

T13.25(a) - (c) are simple particular facts. From (d) every number greater than one is divided by some prime (which may or may not be itself). From (e), a and b are relatively prime iff there is no prime that divides them both; in one direction this is obvious — if a prime divides them both, then they are not relatively prime; in the other direction, if some number other than (the successor of) zero divides them both, then some prime of it divides them both. (f) and (g) let you manipulate G; they are required for (h) which is in turn required for (i) — according to which if Sa is prime and (the successor of) a divides $b \times c$ then (the successor of) a divides b or c; if

Sa is prime and divides $b \times c$ then it must appear in the factorization of b or the factorization of c — so that it divides one or the other.

Now *least common multiple*. Given a function $m(i)$, $lcm\{m(i) \mid i < k\}$ is the least $y > \emptyset$ such that for any $i < k$, $Sm(i)$ divides y. We avoid worries about the case when $m(i) = \emptyset$ by our usual account of factor. And since $y > \emptyset$ it is possible to define a predecessor to the least common multiple, helpful when switching between the numerator and denominator of fractions.

**Def*[*lcm*] $lcm\{m(i) \mid i < k\} = \mu v[v > \emptyset \wedge (\forall i < k)m(i)|v]$

(i) PA $\vdash \exists x[x > \emptyset \wedge (\forall i < k)m(i)|x]$

Hint: This is an argument by IN on k. For the basis, you may assert that $\overline{1} > \emptyset$; then the argument is trivial. For the main argument, under the assumptions $\exists x[x > \emptyset \wedge (\forall i < j)m(i)|x]$ for $\to$I and $a > \emptyset \wedge (\forall i < j)m(i)|a$ for $\exists$E, set out to show $a \times Sm(j) > \emptyset \wedge (\forall i < Sj)m(i)|(a \times Sm(j))$ and generalize.

Because *lcm* is defined by minimization, only the existence condition is required. As a matter of notation, let $l[m]_k = lcm\{m(i) \mid i < k\}$ and, where m is understood, let $l_k = lcm\{m(i) : i < k\}$.

Def[*plm*] $v = plm\{m(i) \mid i < k\} \leftrightarrow Sv = lcm\{m(i) \mid i < k\}$

(i) PA $\vdash \exists v(Sv = l_k)$

(ii) PA $\vdash \forall x \forall y[(Sx = l_k \wedge Sy = l_k) \to x = y]$

Again, let $p[m]_k = plm\{m(i) \mid i < k\}$ and, where m is understood, $p_k = plm\{m(i) \mid i < k\}$.

*T13.26. The following result in PA:

(a) PA $\vdash l_\emptyset = \overline{1}$

(b) PA $\vdash j < k \to m(j)|l_k$

*(c) PA $\vdash (\forall i < k)m(i)|x \to p_k|x$

*(d) PA $\vdash \forall n[(Pr(Sn) \wedge n|l_k) \to (\exists i < k)n|Sm(i)]$

Hints. (c): Let $q = qt(x, p_k)$ and $r = rm(x, p_k)$; assume $(\forall i < k)m(i)|x$ for $\to$I; you have $(\forall y < l_k)\sim[y > \emptyset \wedge (\forall i < k)m(i)|y]$ from def l_k with

T13.19c; you should be able to apply this to show that $r = \emptyset$ and so that $p_k|x$. (d): This is an induction on k. The basis is straightforward given $l_\emptyset = \overline{1}$ from T13.26a; for the main argument, you have $(\forall i < j)m(i)|l_j$ from def l_j; under assumptions $\forall n[(Pr(Sn) \wedge n|l_j) \rightarrow (\exists i < j)n|Sm(i)]$ and $Pr(Sa) \wedge a|l_{Sj}$ for $\rightarrow$I, you should be able to use T13.26c to show $p_{Sj}|(l_j \times Sm(j))$; and from this $a|l_j \vee a|Sm(j)$; in either case, you have your result.

(a) for any function $m(i)$, the least common multiple for $i < 0$ defaults to $\overline{1}$. (b) applies the definition for the result that when $j < k$, $m(j)$ divides $lcm\{m(i)\,|\,i < k\}$. (c) is perhaps best conceived by prime factorization: the least common multiple of some collection has all the primes of its members and no more; but any number into which all the members of the collection divide must include all those primes; so the least common multiple divides it as well. (d) is the related result that if a prime divides the least common multiple of some collection, then it divides some member of the collection.

Finally we arrive at the Chinese Remainder Theorem. As one might expect, this is fundamental to the result we want. Let $m(i)$ be a function whose values are relatively prime — ultimately to be constructed as part of the beta function; $h(i)$ is the function whose values are to be matched by remainders. Then the theorem tells us that if for all $i < k$, $m(i) > \emptyset$ and $m(i) \geq h(i)$, and if for all $i < j < k$, $Rp(Sm(i), Sm(j))$, then $\exists p(\forall i < k)rm(p, m(i)) = h(i)$. This will be the p that figures in the recursion clause.

*T13.27. PA $\vdash [(\forall i < k)(m(i) > \emptyset \wedge m(i) \geq h(i)) \wedge \forall i \forall j(i < j \wedge j < k \rightarrow Rp(Sm(i), Sm(j)))] \rightarrow \exists p(\forall i < k)rm(p, m(i)) = h(i)$ (*CRT*).

Let $\mathcal{A}(k) =_{\text{def}} (\forall i < k)(m(i) > \emptyset \wedge m(i) \geq h(i)) \wedge \forall i \forall j(i < j \wedge j < k \rightarrow Rp(Sm(i), Sm(j)))$

and $\mathcal{B}(k) =_{\text{def}} \exists p(\forall i < k)rm(p, m(i)) = h(i)$.

So we want PA $\vdash \mathcal{A}(k) \rightarrow \mathcal{B}(k)$. By induction on n we show $(\forall n \leq k)(\mathcal{A}(n) \rightarrow \mathcal{B}(n))$. The result follows immediately with $k \leq k$. Here is the overall structure of the argument:

1.	$\emptyset \leq k \to (\mathcal{A}(\emptyset) \to \mathcal{B}(\emptyset))$	[a]
2.	$a \leq k \to (\mathcal{A}(a) \to \mathcal{B}(a))$	A (g →I)
3.	$Sa \leq k$	A (g →I)
4.	$a < k$	3 T13.13k
5.	$a \leq k$	4 T13.13l
6.	$\mathcal{A}(a) \to \mathcal{B}(a)$	2,5 →E
7.	$\mathcal{A}(Sa)$	A (g →I)
8.	$[(\forall i < a)(m(i) > \emptyset \wedge m(i) \geq h(i)) \wedge \forall i \forall j((i < j \wedge j < a) \to \mathcal{R}p(Sm(i), Sm(j)))] \to \exists p(\forall i < a)\mathcal{rm}(p, m(i)) = h(i)$	6 abv
9.	$(\forall i < Sa)(m(i) > \emptyset \wedge m(i) \geq h(i)) \wedge \forall i \forall j((i < j \wedge j < Sa) \to \mathcal{R}p(Sm(i), Sm(j)))$	7 abv
10.	$(\forall i < Sa)(m(i) > \emptyset \wedge m(i) \geq h(i))$	9 ∧E
11.	$\forall i \forall j((i < j \wedge j < Sa) \to \mathcal{R}p(Sm(i), Sm(j)))$	9 ∧E
12.	$\exists p(\forall i < a)\mathcal{rm}(p, m(i)) = h(i)$	[b]
13.	$(\forall i < a)\mathcal{rm}(r, m(i)) = h(i)$	A (g 12∃E)
14.	$\mathcal{R}p(l[m]_a, Sm(a))$	[c]
15.	$Sm(a) > \emptyset$	T13.13e
16.	$l_a > \emptyset$	def l_a
17.	$G(l_a, Sm(a), \overline{1})$	14,15,16 T13.25h
18.	$G(l_a, Sm(a), r + (l_a \dot{-} \overline{1}) \times h(a))$	17 T13.25f
19.	$\exists x \exists y(l_a \times x + [r + (l_a \dot{-} \overline{1}) \times h(a)] = Sm(a) \times y)$	18 def G
20.	$l_a \times b + [r + (l_a \dot{-} \overline{1}) \times h(a)] = Sm(a) \times c$	A (g 19∃E)
21.	$s = l_a \times (b + h(a)) + r$	def
22.	$s = Sm(a) \times c + h(a)$	[d]
23.	$(\forall i < Sa)\mathcal{rm}(s, m(i)) = h(i)$	[e]
24.	$\exists p(\forall i < Sa)\mathcal{rm}(p, m(i)) = h(i)$	23 ∃I
25.	$\mathcal{B}(Sa)$	24 abv
26.	$\mathcal{B}(Sa)$	19,20-25 ∃E
27.	$\mathcal{B}(Sa)$	12,13-26 ∃E
28.	$\mathcal{A}(Sa) \to \mathcal{B}(Sa)$	7-27 →I
29.	$Sa \leq k \to (\mathcal{A}(Sa) \to \mathcal{B}(Sa))$	3-28 →I
30.	$[a \leq k \to (\mathcal{A}(a) \to \mathcal{B}(a))] \to [Sa \leq k \to (\mathcal{A}(Sa) \to \mathcal{B}(Sa))]$	2-29 →I
31.	$\forall n([n \leq k \to (\mathcal{A}(n) \to \mathcal{B}(n))] \to [Sn \leq k \to (\mathcal{A}(Sn) \to \mathcal{B}(Sn))])$	30 ∀I
32.	$(\forall n \leq k)(\mathcal{A}(n) \to \mathcal{B}(n))$	1,31 IN
33.	$k \leq k$	T13.13l
34.	$\mathcal{A}(k) \to \mathcal{B}(k)$	32,33 (∀E)

Hints. (c): Suppose otherwise; with T13.25e there is a u such that $Pr(Su) \wedge u|l_a \wedge u|Sm(a)$; then with T13.26d there is a $v < a$ such that $u|Sm(v)$ so that with (11) $\mathcal{R}p(Sm(v), Sm(a))$. But this is impossible with $u|Sm(a)$, $u|Sm(v)$ and T13.25e. (d): By *Def*[*lcm*], $l_a > \emptyset$ so that $h(a)l_a > h(a)$. Then with T13.23a and T13.23n you can show, $s = (l_a \times b + [r + (l_a \dot{-} \overline{1}) \times h(a)]) + h(a)$ and apply (20). (e): Suppose for (∀I) $u < Sa$; then $u < a \vee u = a$. In the first case, with T13.26b and T13.24d $m(u)|l_a(b + h(a))$; so that there is a v such that $Sm(u)v = l_a(b + h(a))$; then using (21) and T13.24k, $\mathcal{rm}(d, m(u)) =$

$rm(s, m(u))$; so that you can apply (13). In the second case, with (22) and T13.24k $rm(d, m(u)) = rm(h(u), m(u))$; but from (10), $m(u) \geq h(u)$ and you will be able to show that $rm(h(u), m(u)) = h(u)$.

(12) is simple enough once you use (10) and (11) to generate the antecedent to (8). After that, we expect (14) insofar as the values of $Sm(i)$ are relatively prime up to and including a; so the values of $Sm(i)$ have no primes in common; since l_a includes just the primes of members $< a$, it has no prime in common with $Sm(a)$; so l_a and $Sm(a)$ are relatively prime. This yields a straight path to (20). Then the idea is that s appears in the forms from both (21) and (22). From the version on (21), for any $i < a$, the remainder of $m(i)$ and s is the same as the remainder of $m(i)$ with r — that is $h(i)$, since $m(i)$ divides the first term evenly. And from the version on (22), the remainder of $m(a)$ and s is equal to $h(a)$ — since $m(a)$ divides the first term evenly and $m(a) \geq h(a)$. Putting these together, for any $i < Sa$, the remainder of $m(i)$ and s is $h(i)$. The "trick" is in the construction of s (following Boolos, *The Logic of Provability*, 30-31).

For our final results, we require a couple notions for maximum value. First *maxs* for the maximum from a *set* of values, and then *maxp* for the greatest of a *pair*.

**Def*[*maxs*] PA $\vdash v = maxs\{m(i) \mid i < k\} \leftrightarrow (k = \emptyset \wedge v = \emptyset) \vee ((\exists i < k)m(i) = v \wedge (\forall i < k)m(i) \leq v)$

Let $\mathcal{A}(k, v) =_{\text{def}} (k = \emptyset \wedge v = \emptyset)$ and $\mathcal{B}(k, v) =_{\text{def}} (\exists i < k)m(i) = v \wedge (\forall i < k)m(i) \leq v$. Then we require,

(i) PA $\vdash \exists v[\mathcal{A}(k, v) \vee \mathcal{B}(k, v)]$

(ii) PA $\vdash \forall y \forall z[([\mathcal{A}(k, y) \vee \mathcal{B}(k, y)] \wedge [\mathcal{A}(k, z) \vee \mathcal{B}(k, z)]) \rightarrow y = z]$

The argument for (ii) is long and disjunctive, but straightforward. (i) is an argument by IN on k. It is not difficult, but, again, long and disjunctive. Here is the basic structure including key subgoals.

1.	$\mathcal{A}(\emptyset,\emptyset) \vee \mathcal{B}(\emptyset,\emptyset)$	[a]
2.	$\exists v[\mathcal{A}(\emptyset,v) \vee \mathcal{B}(\emptyset,v)]$	1∃I
3.	$\exists v[\mathcal{A}(j,v) \vee \mathcal{B}(j,v)]$	A (g →I)
4.	$\mathcal{A}(j,u) \vee \mathcal{B}(j,u)$	A (g 3∃E)
5.	$j = \emptyset \vee j \neq \emptyset$	T3.1
6.	$j = \emptyset$	A (g 5∨E)
7.	$\mathcal{A}(Sj,m(\emptyset)) \vee \mathcal{B}(Sj,m(\emptyset))$	[b]
8.	$\exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	7 ∃I
9.	$j \neq \emptyset$	A (g 5∨E)
10.	$j \neq \emptyset \vee u \neq \emptyset$	9 ∨I
11.	$\sim(j = \emptyset \wedge u = \emptyset)$	10 DeM
12.	$\sim\mathcal{A}(j,u)$	11 abv
13.	$\mathcal{B}(j,u)$	4,12 DS
14.	$(\exists i < j)m(i) = u \wedge (\forall i < j)m(i) \leq u$	13 abv
15.	$(\forall i < j)m(i) \leq u$	14 ∧E
16.	$(\exists i < j)m(i) = u$	14 ∧E
17.	$m(a) = u$	A (g 16(∃E))
18.	$a < j$	
19.	$m(j) \leq m(a) \vee m(j) > m(a)$	T13.13p
20.	$m(j) \leq m(a)$	A (g 19∨E)
21.	$\mathcal{A}(Sj,m(a)) \vee \mathcal{B}(Sj,m(a))$	[c]
22.	$\exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	21 ∃I
23.	$m(j) > m(a)$	A (g 19∨E)
24.	$\mathcal{A}(Sj,m(j)) \vee \mathcal{B}(Sj,m(j))$	[d]
25.	$\exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	24 ∃I
26.	$\exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	19,20-22,23-25 ∨E
27.	$\exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	16,17-26 (∃E)
28.	$\exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	5,6-8,9-27 ∨E
29.	$\exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	3,4-28 ∃E
30.	$\exists v[\mathcal{A}(j,v) \vee \mathcal{B}(j,v)] \to \exists v[\mathcal{A}(Sj,v) \vee \mathcal{B}(Sj,v)]$	3-29 →I
31.	$\forall y(\exists v[\mathcal{A}(y,v) \vee \mathcal{B}(y,v)] \to \exists v[\mathcal{A}(Sy,v) \vee \mathcal{B}(Sy,v)])$	30 ∀I
32.	$\exists v[\mathcal{A}(k,v) \vee \mathcal{B}(k,v)]$	2,31 IN

So the generalization is from different individuals in the different cases [a], [b], [c] and [d]. As a matter of notation, let $maxs[m]_k = maxs\{m(i) \mid i < k\}$ and where m is understood, $maxs_k = maxs\{m(i) \mid i < k\}$.

**Def*[*maxp*] PA $\vdash v = maxp(x,y) \leftrightarrow (x \geq y \wedge v = x) \vee (x < y \wedge v = y)$

(i) PA $\vdash \exists v[(x \geq y \wedge v = x) \vee (x < y \wedge v = y)]$

(ii) PA $\vdash \forall u \forall v[([(x \geq y \wedge u = x) \vee (x < y \wedge u = y)] \wedge [(x \geq y \wedge v = x) \vee (x < y \wedge v = y)]) \to u = v]$

And a couple of results that make the obvious applications from the definitions.

*T13.28. The following result in PA.

(a) PA $\vdash maxp(x, y) \geq x \wedge maxp(x, y) \geq y$

(b) PA $\vdash (\forall i < k)m(i) \leq maxs_k$

These simply state the obvious: that the maximum is greater than or equal to the rest. For (a) that the maximum is the greater of the the two in the pair; for (b) that the maximum is the greater of the values of the function.

Now we are in a position to generate some results for the β function. With values of q and $m(i)$ as below, we may demonstrate the antecedent to the *CRT* (T13.27), and so obtain its consequent — with application to to the β-function.

*T13.29. PA $\vdash \exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$.

Let $r =_{\text{def}} maxp(k, maxs[h]_k)$;

$s =_{\text{def}} Sr$;

$q =_{\text{def}} lcm\{i \mid i < s\}$;

$m(i) =_{\text{def}} q \times Si$.

Then $\beta(p, q, i) = rm(p, q \times Si)$. And we may reason,

1.	$(\forall i < k)(m(i) > \emptyset \wedge m(i) \geq h(i))$	[i]
2.	$\forall i \forall j[(i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j))]$	[ii]
3.	$\exists p(\forall i < k)rm(p, m(i)) = h(i)$	1,2 T13.27
4.	$m(i) = q \times Si$	def
5.	$\exists p(\forall i < k)rm(p, q \times Si) = h(i)$	3,4 =E
6.	$\beta(p, q, i) = rm(p, q \times Si)$	def
7.	$\exists p(\forall i < k)\beta(p, q, i) = h(i)$	5,6 =E
8.	$\quad(\forall i < k)\beta(p, q, i) = h(i)$	A (g 7∃E)
9.	$\quad\exists q(\forall i < k)\beta(p, q, i) = h(i)$	8 ∃I
10.	$\quad\exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$	9 ∃I
11.	$\exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$	7,8-10 ∃E

So the demonstration reduces to that of (i) and (ii), the two conjuncts to the antecedent of *CRT* (T13.27). (i): Under the assumption $j < k$ for (∀I) it will be easy to show $m(j) > \emptyset$; then you will be able to use T13.28 to show $h(j) < s$; but also with T13.26b that $r|q$ and from this that $s \leq q$ which gives $s \leq q \times Sj$ and the result you want. (ii): Here is the main outline of the argument.

1.	$i < j \wedge j < k$	A g $\rightarrow$I
2.	$i < j$	1 $\wedge$E
3.	$j < k$	1 $\wedge$E
4.	$\sim Rp(Sm(i), Sm(j))$	A (c $\sim$I)
5.	$\exists x[Pr(Sx) \wedge x \vert S(q \times Si) \wedge x \vert S(q \times Sj)]$	4 T13.25e
6.	$Pr(Sa) \wedge a \vert S(q \times Si) \wedge a \vert S(q \times Sj)$	A (c 5$\exists$E)
7.	$Pr(Sa)$	6 $\wedge$E
8.	$a \vert S(q \times Si)$	6 $\wedge$E
9.	$a \vert S(q \times Sj)$	6 $\wedge$E
10.	$a \vert q(j \dot{-} i)$	[a]
11.	$a \vert q \vee a \vert (j \dot{-} i)$	7,10 T13.25i
12.	$a \vert q$	A (g 11$\vee$E)
13.	$a \vert q$	12 R
14.	$a \vert (j \dot{-} i)$	A (g 11$\vee$E)
15.	$a \vert q$	[b]
16.	$a \vert q$	11,12-13,14-15 $\vee$E
17.	$a \vert (q \times Si)$	16 T13.24d
18.	$S(q \times Si) > q \times Si$	T13.13g
19.	$S(q \times Si) \geq q \times Si$	18 T13.13l
20.	$a \vert (S(q \times Si) \dot{-} (q \times Si))$	19,8,17 T13.24h
21.	$a \vert \overline{1}$	20 T13.23f
22.	$S\emptyset < Sa$	def Pr
23.	$\emptyset < a$	22 T13.13j
24.	$a \nmid \overline{1}$	23 T13.24i
25.	$\perp$	21,24 $\perp$I
26.	$\perp$	5,6-25 $\exists$E
27.	$Rp(Sm(i), Sm(j))$	4-26 $\sim$E
28.	$(i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j))]$	1-27 $\rightarrow$I
29.	$\forall i \forall j[(i < j \wedge j < k) \rightarrow Rp(Sm(i), Sm(j))]$	28 $\forall$I

Hints. (a): With $i < j$ you will be able to show $a|(S(q \times Sj) \dot{-} S(q \times Si))$; and with some work that $S(q \times Sj) \dot{-} S(q \times Si) = q(j \dot{-} i)$. (b): With $i < j$, you have $j \dot{-} i > \emptyset$; so there is an l such that $Sl + \emptyset = j \dot{-} i$; you will be able to show $a|Sl$ and with T13.26b, $l|q$ so with T13.24f, $a|q$.

Next a theorem that leads directly to our main result. We show that given $\beta(r, s, i)$ there are sure to be p and q such that $\beta(p, q, i)$ is like $\beta(r, s, i)$ for $i < k$ and for arbitrary n, $\beta(p, q, k) = n$. This is because we may *define* a function h which is like $\beta(r, s, i)$ for $i < k$ and otherwise n — and find p, q such that $\beta(p, q, i)$ matches it. As a preliminary,

Def$[h(i)]$ PA $\vdash v = h(i) \leftrightarrow (i < k \wedge v = \beta(r, s, i)) \vee i \geq k \wedge v = n)$

(i) PA $\vdash \exists v[(i < k \wedge v = \beta(r, s, i)) \vee i \geq k \wedge v = n)]$

(ii) PA $\vdash \forall x \forall y[([(i < k \wedge x = \beta(r,s,i)) \vee i \geq k \wedge x = n)] \wedge [(i < k \wedge y = \beta(r,s,i)) \vee i \geq k \wedge y = n)]) \rightarrow x = y]$

Then,

*T13.30. PA $\vdash \exists p \exists q[(\forall i < k)\beta(p,q,i) = \beta(r,s,i) \wedge \beta(p,q,k) = n]$.

Hints: From *Def*$[h(i)]$ you have $(k < k \wedge h(k) = \beta(r,s,k)) \vee (k \geq k \wedge h(k) = n)$ and $(l < k \wedge h(l) = \beta(r,s,l)) \vee (l \geq k \wedge h(l) = n)$; and from T13.29 applied to Sk, $\exists p \exists q(\forall i < Sk)\beta(p,q,i) = h(i)$; then with $(\forall i < Sk)\beta(a,b,i) = h(i)$ for $\exists$E, you will be able to show that $\beta(a,b,k) = n$ and under $l < k$ for ($\forall$I) that $\beta(a,b,l) = \beta(r,s,l)$.

For application of this theorem, it is important that free variables are universally quantified. So the theorem is effectively, $\forall k \forall n \forall r \forall s \exists p \exists q[(\forall i < k)\beta(p,q,i) = \beta(r,s,i) \wedge \beta(p,q,k) = n]$

And finally the result we have been after in this section: As before, let $\mathscr{F}(\vec{x}, y, v)$ be our formula,

$$\exists p \exists q[\beta(p,q,\emptyset) = \mathbb{g}(\vec{x}) \wedge (\forall i < y)\mathbb{h}(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = z]$$

Then we want, PA $\vdash \exists v \mathscr{F}(\vec{x}, y, v)$.

*T13.31. PA $\vdash \exists v \exists p \exists q[\beta(p,q,\emptyset) = g(\vec{x}) \wedge (\forall i < y)h(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = v]$.

Let $\mathscr{F}(\vec{x}, y, v)$ be as above; the argument is by IN on y. The zero case is left as an exercise. Here is the main argument.

1.	$\exists v\mathscr{F}(\vec{x}, \emptyset, v)$	zero case
2.	$\exists v\mathscr{F}(\vec{x}, j, v)$	A (g $\rightarrow$I)
3.	$\exists v\exists p\exists q[\beta(p,q,\emptyset) = g(\vec{x}) \wedge (\forall i < j)h(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,j) = v]$	2 abv
4.	$\beta(a,b,\emptyset) = g(\vec{x}) \wedge (\forall i < j)h(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si) \wedge \beta(a,b,j) = z$	A (g 3$\exists$E)
5.	$\beta(a,b,\emptyset) = g(\vec{x})$	4 $\wedge$E
6.	$(\forall i < j)h(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si)$	4 $\wedge$E
7.	$\exists p\exists q[(\forall i < Sj)\beta(p,q,i) = \beta(a,b,i) \wedge \beta(p,q,Sj) = h(\vec{x}, j, \beta(a,b,j))]$	T13.30 $\forall$E
8.	$(\forall i < Sj)\beta(c,d,i) = \beta(a,b,i) \wedge \beta(c,d,Sj) = h(\vec{x}, j, \beta(a,b,j))$	A (g 7$\exists$E)
9.	$(\forall i < Sj)\beta(c,d,i) = \beta(a,b,i)$	8 $\wedge$E
10.	$\beta(c,d,Sj) = h(\vec{x}, j, \beta(a,b,j))$	8 $\wedge$E
11.	$\emptyset < Sj$	T13.13e
12.	$\beta(c,d,\emptyset) = \beta(a,b,\emptyset)$	9,11 ($\forall$E)
13.	$\beta(c,d,\emptyset) = g(\vec{x})$	5,12 =E
14.	$l < Sj$	A (g ($\forall$I))
15.	$\beta(c,d,l) = \beta(a,b,l)$	9,14 ($\forall$E)
16.	$l < j \vee l = j$	14 T13.13m
17.	$l < j$	A (g 16$\vee$E)
18.	$h(\vec{x}, l, \beta(a,b,l)) = \beta(a,b,Sl)$	6,17 ($\forall$E)
19.	$Sl < Sj$	17 T13.13j
20.	$\beta(c,d,Sl) = \beta(a,b,Sl)$	9,19 $\forall$E
21.	$h(\vec{x}, l, \beta(a,b,l)) = \beta(c,d,Sl)$	18,20 =E
22.	$l = j$	A (g 16$\vee$E)
23.	$h(\vec{x}, l, \beta(a,b,l)) = \beta(c,d,Sl)$	10,22 =E
24.	$h(\vec{x}, l, \beta(c,d,l)) = \beta(c,d,Sl)$	15,23 =E
25.	$h(\vec{x}, l, \beta(c,d,l)) = \beta(c,d,Sl)$	16,17-21,22-24 $\vee$E
26.	$(\forall i < Sj)h(\vec{x}, i, \beta(c,d,i)) = \beta(c,d,Si)$	14-25 ($\forall$I)
27.	$\beta(c,d,Sj) = \beta(c,d,Sj)$	=I
28.	$\beta(c,d,\emptyset) = g(\vec{x}) \wedge (\forall i < Sj)h(\vec{x}, i, \beta(c,d,i)) = \beta(c,d,Si) \wedge \beta(c,d,Sj) = \beta(c,d,Sj)$	13,26,27 $\wedge$I
29.	$\exists v\exists p\exists q[\beta(p,q,\emptyset) = g(\vec{x}) \wedge (\forall i < Sj)h(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,Sj) = v]$	28 $\exists$I
30.	$\exists v\mathscr{F}(\vec{x}, Sj, v)$	29 abv
31.	$\exists v\mathscr{F}(\vec{x}, Sj, v)$	7,8-30 $\exists$E
32.	$\exists v\mathscr{F}(\vec{x}, Sj, v)$	3,4-31 $\exists$E
33.	$\exists v\mathscr{F}(\vec{x}, j, v) \rightarrow \exists v\mathscr{F}(\vec{x}, Sj, v)$	2-32 $\rightarrow$I
34.	$\forall y[\exists v\mathscr{F}(\vec{x}, y, v) \rightarrow \exists v\mathscr{F}(\vec{x}, Sy, v)]$	33 $\forall$I
35.	$\exists v\mathscr{F}(\vec{x}, y, v)$	1,34 IN

From the assumption, there are a, b such that the β-function has the right features for every $i < j$. With T13.30 there are c, d such that the β-function has the right features for $i < Sj$. The derivation establishes that this is so and generalizes.

This completes the demonstration of T13.21. So for any friendly recursive function $\mathsf{r}(\vec{\mathsf{x}})$ and original formula $\mathscr{R}(\vec{x}, v)$ by which it is expressed and captured, PA defines a function $\mathscr{r}(\vec{x})$ such that PA $\vdash v = \mathscr{r}(\vec{x}) \leftrightarrow \mathscr{R}(\vec{x}, v)$. In particular, then, PA

defines functions corresponding to all the primitive recursive functions from chapter 12.

In addition, say a recursive relation is *friendly* iff it has a friendly characteristic function. Then as a simple corollary, PA defines relations corresponding to each friendly recursive relation, equivalent to the original formulas used to express them.

T13.32. For any friendly recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ with characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$, PA defines a relation $\mathbb{R}(\vec{x})$ such that PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$. As a simple corollary, where $\mathsf{R}(\vec{\mathsf{x}})$ is originally captured by $\mathcal{R}(\vec{x}, \emptyset)$, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, \emptyset)$.

Suppose a friendly recursive relation R has recursive characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$. Since R is friendly, it has a friendly characteristic function that is defined in PA. Set,

$$\text{PA} \vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$$

Then PA defines $\mathbb{R}(\vec{x})$. In fact, for relations defined in chapter 12, we will want to define relations whose structure matches the structure of functions there defined. For this, it will be helpful to obtain the same result by an (informal) induction.

(a) Say an *atomic* recursive relation is one like EQ, LEQ or LESS whose characteristic function does not depend on the characteristic functions of other recursive relations. Then let,

$$\text{PA} \vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathscr{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$$

(b) Now suppose PA $\vdash \mathbb{P}_1(\vec{x}) \leftrightarrow \mathscr{ch}_{\mathsf{P}_1}(\vec{x}) = \emptyset$ and ...and PA $\vdash \mathbb{P}_n(\vec{x}) \leftrightarrow \mathscr{ch}_{\mathsf{P}_n}(\vec{x}) = \emptyset$. And consider a recursive operator, $\mathsf{OP}(\mathsf{P}_1(\vec{\mathsf{x}}) \dots \mathsf{P}_n(\vec{\mathsf{x}}))$ with characteristic function $\mathsf{f}(\mathsf{ch}_{\mathsf{P}_1}(\vec{\mathsf{x}}) \dots \mathsf{ch}_{\mathsf{P}_n}(\vec{\mathsf{x}}))$. Since $\mathsf{f}(\mathsf{ch}_{\mathsf{P}_1}(\vec{\mathsf{x}}) \dots \mathsf{ch}_{\mathsf{P}_n}(\vec{\mathsf{x}}))$ is friendly, PA defines $\mathscr{f}(\vec{x})$. Let $\mathscr{c}_{\mathsf{P}}(\vec{x}) = \mu v[(\mathbb{P}(\vec{x}) \wedge v = \emptyset) \vee (\sim\mathbb{P}(\vec{x}) \wedge v = \overline{1})]$ and set,

$$\text{PA} \vdash \mathbb{O}p(\mathbb{P}_1(\vec{x}) \dots \mathbb{P}_n(\vec{x})) \leftrightarrow \mathscr{f}(\mathscr{c}_{\mathsf{P}_1}(\vec{x}) \dots \mathscr{c}_{\mathsf{P}_n}(\vec{x})) = \emptyset$$

Officially, from subsection 13.3.1 we do not *define* new operators into the language; rather, the new operator applied to $\mathbb{P}_1 \dots \mathbb{P}_n$ abbreviates an expression of which $\mathbb{P}_1 \dots \mathbb{P}_n$ are parts. But by T13.37 (which we shall see shortly),

PA $\vdash \mathbb{ch}_{\mathsf{P}}(\vec{x}) = \emptyset \vee \mathbb{ch}_{\mathsf{P}}(\vec{x}) = \overline{1}$; and it is easy to see, PA $\vdash \mathbb{c}_{\mathsf{P}}(\vec{x}) = \mathbb{ch}_{\mathsf{P}}(\vec{x})$; so that PA $\vdash \mathbb{Op}(\mathbb{P}_1(\vec{x}) \ldots \mathbb{P}_n(\vec{x})) \leftrightarrow \mathbb{f}(\mathbb{ch}_{\mathsf{P}_1}(\vec{x}) \ldots \mathbb{ch}_{\mathsf{P}_n}(\vec{x})) = \emptyset$. Now for any $\mathsf{R}(\vec{\mathsf{x}}) = \mathsf{OP}(\mathsf{P}_1(\vec{\mathsf{x}}) \ldots \mathsf{P}_n(\vec{\mathsf{x}}))$ set,

$$\text{PA} \vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{Op}(\mathbb{P}_1(\vec{x}) \ldots \mathbb{P}_n(\vec{x}))$$

Then PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{f}(\mathbb{ch}_{\mathsf{P}_1}(\vec{x}) \ldots \mathbb{ch}_{\mathsf{P}_n}(\vec{x})) = \emptyset$; which is to say, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$.

(d) So for any primitive recursive relation defined in chapter 12, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$. Further, with T13.21, PA $\vdash v = \mathbb{ch}_{\mathsf{R}}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, v)$; so PA $\vdash \emptyset = \mathbb{ch}_{\mathsf{R}}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, \emptyset)$; so PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, \emptyset)$.

From part (a) we have, say, PA $\vdash \mathbb{Eq}(\vec{x}) \leftrightarrow \mathbb{ch}_{\mathsf{EQ}}(\vec{x}) = \emptyset$. As an example for (b), $\mathsf{DSJ}(\mathsf{P}(\vec{\mathsf{x}}), \mathsf{Q}(\vec{\mathsf{x}}))$ has characteristic function $\mathsf{times}(\mathsf{ch}_{\mathsf{P}}(\vec{\mathsf{x}}), \mathsf{ch}_{\mathsf{Q}}(\vec{\mathsf{x}}))$; so we set PA $\vdash \mathbb{Dsj}(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{x})) \leftrightarrow \mathbb{times}(\mathbb{ch}_{\mathsf{P}}(\vec{x}), \mathbb{ch}_{\mathsf{Q}}(\vec{x})) = \emptyset$; then where $\mathsf{R}(\vec{\mathsf{x}}) = \mathsf{DSJ}(\mathsf{P}(\vec{\mathsf{x}}), \mathsf{Q}(\vec{\mathsf{x}}))$, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{Dsj}(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{x}))$.

Thus PA defines both functions and relations corresponding to the friendly recursive functions and relations, equivalent to the original formulas used to express and capture them. As we shall see, these theorems let us "write down" definitions in PA given recursive definitions from before. This is not everything we want. But it is a start.

*E13.12. Show (i) and (ii) for *Def*[$\dot{-}$]. Then show T13.23 (a) and (m). Hard core: show all of the results in T13.23.

*E13.13. Show T13.24d and T13.24i. Hard core: show all of the results in T13.24.

*E13.14. Provide a complete demonstration of T13.25h including the justification for d. Hard core: Show all of the results from T13.25.

*E13.15. Show the condition for *Def*[*lcm*] and provide a demonstration for T13.26d. Hard core: show all of the results for *Def*[*lcm*], *Def*[*plm*] and T13.26.

*E13.16. Provide derivations to show each of [a] - [e] to complete the derivation for T13.27.

Font conventions

At different stages, we employ different fonts for items of different sorts. For the most part, this should have been straightforward. Here we collect them together.

1. Expressions of symbolic object languages are given in italics; these include the function (lowercase) and relation (first letter uppercase) symbols abbreviated or defined in Q and PA.

   function, *Relation*

2. Objects from the semantic account are indicated by a sans-serif font; these include recursive functions (lowercase) and relations (small-caps) — and bold when special symbols are used.

   $\mathsf{function}$, $\mathsf{RELATION}$,

3. The language for description of expressions in the formal object language uses script variables,

   $\mathscr{P}$, $\mathscr{p}$

4. The language for description of metalinguistic expressions uses Fraktur variables,

   $\mathfrak{A}$, $\mathfrak{a}$

5. Function and relation symbols introduced into PA from recursive functions and relations by T13.21 and T13.32 have their first character in a "hollow" blackboard bold font — these are not automatically the equivalent to ones that may be described in (1), though we may set out to demonstrate equivalence.

   $\mathbb{f}unction$, $\mathbb{R}elation$

6. Object expressions for computer languages are given in a typewriter font,

   `Expression`

7. In addition, for informal inductions italic i, j generally index objects arranged in series, but i, j when the objects are specifically the members of $\mathbb{N}$.

*E13.17. Complete the argument for condition (i) of *Def*[*maxs*] for by producing arguments for (a), (b), (c) and (d). Hard core: Provide complete justifications for *Def*[*maxs*] and *Def*[*maxp*]; and show each of the results in T13.28.

*E13.18. Complete the demonstration for T13.29.

*E13.19. Show T13.30. Hard core: show the conditions for *Def*[$h(i)$].

*E13.20. Complete the demonstration of T13.31 by showing the zero case.

E13.21. Give the demonstration to show PA $\vdash \mathbb{O}p(\mathbb{P}_1(\vec{x}) \dots \mathbb{P}_n(\vec{x})) \leftrightarrow \mathbb{f}(\mathbb{ch}_{\mathsf{P}_1}(\vec{x}) \dots \mathbb{ch}_{\mathsf{P}_n}(\vec{x})) = \emptyset$ from (b) of T13.32.

13.4 The Second Condition: $\Box(\mathcal{P} \to \mathcal{Q}) \to (\Box\mathcal{P} \to \Box\mathcal{Q})$

We turn now to demonstration of the second derivability condition. Again there is some background — after which demonstration of the condition itself is straightforward. The idea is simple: Suppose both $\Box(\mathcal{P} \to \mathcal{Q})$ and $\Box\mathcal{P}$. Then there are j and k such that $\textsf{PRFT}(\mathsf{j}, \ulcorner\mathcal{P} \to \mathcal{Q}\urcorner)$ and $\textsf{PRFT}(\mathsf{k}, \ulcorner\mathcal{P}\urcorner)$. Intuitively, then, $\mathsf{l} = \mathsf{j} \star \mathsf{k} \star 2^{\ulcorner\mathcal{Q}\urcorner}$ numbers a proof of $\mathcal{Q}$ — for we prove $\mathcal{P} \to \mathcal{Q}$, then $\mathcal{P}$, then $\mathcal{Q}$ follows immediately as the last line by MP. So the idea is that $\textsf{PRFT}(\mathsf{l}, \ulcorner\mathcal{Q}\urcorner)$ so that $\Box\mathcal{Q}$ follows from the assumptions. The task is to prove all of this in PA.

13.4.1 Some Applications

We have now shown that PA defines all the functions we require. However, this is not everything we want. Observe that $\mathbb{plus}(x, y)$, say, is defined by a complex expression through recursion, and so is not the same expression as our old friend $x + y$. Thus it is not obvious that our standard means for manipulation of $+$ apply to $\mathbb{plus}$. We could recover our ordinary results if we could show PA $\vdash x + y = \mathbb{plus}(x, y)$. And similar comments apply to other ordinary functions and relations. Thus initially we seek to show that defined relations functions are equivalent to ones with which we are familiar. Again many details are shifted to exercises and/or answers to exercises.

Equivalencies. We begin with equivalences between functions and relations already defined in PA, and those that result from the recursive functions and relations by T13.21 and T13.32. So we begin with functions and relations from $\mathscr{L}_{NT}$ including $S, +, \times, =, \leq, <$, truth functional operators, bounded quantifiers and bounded minimization. Given the way recursive functions are constructed, these will require a few additional notions along the way.

As a preliminary, however, we require a result that is fundamental to every case where a function is defined by recursion. As above let $\mathcal{F}(\vec{x}, y, v)$ be,

$$\exists p \exists q[\beta(p,q,\emptyset) = g(\vec{x}) \wedge (\forall i < y)h(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = z]$$

and suppose PA $\vdash v = f(\vec{x}, y) \leftrightarrow \mathcal{F}(\vec{x}, y, v)$ so that $f(\vec{x}, y)$ is defined by recursion; then the standard recursive conditions apply. That is,

T13.33. Suppose $f(\vec{x}, y)$ is defined by $g(\vec{x})$ and $h(\vec{x}, y, u)$ so that PA $\vdash v = f(\vec{x}, y) \leftrightarrow \mathcal{F}(\vec{x}, y, v)$. Then,

(a) PA $\vdash f(\vec{x}, \emptyset) = g(\vec{x})$

(b) PA $\vdash f(\vec{x}, S(y)) = h(\vec{x}, y, f(\vec{x}, y))$

Hint: (a) follows easily in 6 lines with $\exists p \exists q[\beta(p,q,\emptyset) = g(\vec{x}) \wedge (\forall i < \emptyset)h(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,\emptyset) = f(\vec{x}, \emptyset)]$. For (b),

1.	$\exists p \exists q[\beta(p,q,\emptyset) = g(\vec{x}) \wedge (\forall i < Sy)h(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,Sy) = f(\vec{x}, Sy)]$	def
2.	$\beta(a,b,\emptyset) = g(\vec{x}) \wedge (\forall i < Sy)h(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si) \wedge \beta(a,b,Sy) = f(\vec{x}, Sy)$	A (g 1$\exists$E)
3.	$(\forall i < Sy)h(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si)$	2 $\wedge$E
4.	$y < Sy$	T13.13g
5.	$h(\vec{x}, y, \beta(a,b,y)) = \beta(a,b,Sy)$	3,4 ($\forall$E)
6.	$\beta(a,b,Sy) = f(\vec{x}, Sy)$	2 $\wedge$E
7.	$f(\vec{x}, Sy) = h(\vec{x}, y, \beta(a,b,y))$	5,6 =E
8.	$\beta(a,b,\emptyset) = g(\vec{x})$	2 $\wedge$E
9.	$j < y$	A (g ($\forall$I))
10.	$j < Sy$	9 and T13.13g
11.	$h(\vec{x}, j, \beta(a,b,j)) = \beta(a,b,Sj)$	3,10 ($\forall$E)
12.	$(\forall i < y)h(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si)$	9-11 ($\forall$I)
13.	$\beta(a,b,y) = \beta(a,b,y)$	=I
14.	$\beta(a,b,\emptyset) = g(\vec{x}) \wedge (\forall i < y)h(\vec{x}, i, \beta(a,b,i)) = \beta(a,b,Si) \wedge \beta(a,b,y) = \beta(a,b,y)$	8,12,13 $\wedge$I
15.	$\exists p \exists q[\beta(p,q,\emptyset) = g(\vec{x}) \wedge (\forall i < y)h(\vec{x}, i, \beta(p,q,i)) = \beta(p,q,Si) \wedge \beta(p,q,y) = \beta(a,b,y)]$	14 $\exists$I
16.	$f(\vec{x}, y) = \beta(a,b,y)$	15 def
17.	$f(\vec{x}, Sy) = h(\vec{x}, y, f(\vec{x}, y))$	7,16 =E
18.	$f(\vec{x}, S(y)) = h(\vec{x}, y, f(\vec{x}, y))$	1,2-17 $\exists$E

The key stages of this argument are at (7) which has the result with $\beta(a,b,y)$ where we want $f(\vec{x}, y)$ and then (16) which shows they are one and the same.

From this theorem, our defined functions behave like ones we have seen before, with clauses for the basis and then for successor. This lets us manipulate the functions very much as before. The importance of this point will emerge shortly, in application to recursive cases.

Observe that from T13.21 PA proves results "parallel" to friendly recursive definitions. From the basis, PA defines $\mathbb{suc}$, $\mathbb{zero}$ and $\mathbb{idnt}$. Then when $\mathsf{f}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}}) = \mathsf{h}(\vec{\mathsf{x}}, \mathsf{g}(\vec{\mathsf{y}}), \vec{\mathsf{z}})$ by composition, PA $\vdash \mathbb{f}(\vec{x}, \vec{y}, \vec{z}) = \mathbb{h}(\vec{x}, \mathbb{g}(\vec{y}), \vec{z})$. If $\mathsf{f}(\vec{\mathsf{x}}) = \mu\mathsf{y}[\mathsf{g}(\vec{\mathsf{x}}, \mathsf{y})]$ by friendly regular minimization, PA $\vdash \mathbb{f}(\vec{x}) = \mu y[\mathbb{g}(\vec{x}, y)]$. And with T13.33, when $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{y})$ is defined by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$, then PA $\vdash \mathbb{f}(\vec{x}, \emptyset) = \mathbb{g}(\vec{x})$ and PA $\vdash \mathbb{f}(\vec{x}, Sy) = \mathbb{h}(\vec{x}, y, \mathbb{f}(\vec{x}, y))$. In addition, we have here an additional mode of definition *within* PA. For any defined $\mathbb{g}(\vec{x})$ and $\mathbb{h}(\vec{x}, y, u)$ there is always a corresponding recursive $\mathsf{f}(\vec{\mathsf{x}})$; thus there is a defined $\mathbb{f}(\vec{x})$ such that PA $\vdash \mathbb{f}(\vec{x}, \emptyset) = \mathbb{g}(\vec{x})$ and PA $\vdash \mathbb{f}(\vec{x}, Sy) = \mathbb{h}(\vec{x}, y, \mathbb{f}(\vec{x}, y))$.

And with T13.32 a similar point applies to friendly recursive relations. There are $\mathbb{Eq}$, $\mathbb{Leq}$ and $\mathbb{Less}$. Then for any $\mathsf{R}(\vec{\mathsf{x}}) = \mathsf{OP}(\mathsf{P}_1(\vec{\mathsf{x}}) \dots \mathsf{P}_n(\vec{\mathsf{x}}))$, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{Op}(\mathbb{P}_1(\vec{x}) \dots \mathbb{P}_n(\vec{x}))$. This lets us "write down" defined functions and relations directly from the recursive definitions. With this said, we turn to our results.

T13.34. The following result in PA.

(a) PA $\vdash \mathbb{suc}(x) = Sx$

1.	$v = \mathbb{suc}(x) \leftrightarrow Sx = v$	def $\mathbb{suc}$
2.	$\mathbb{suc}(x) = \mathbb{suc}(x) \leftrightarrow Sx = \mathbb{suc}(x)$	1 $\forall$E
3.	$\mathbb{suc}(x) = \mathbb{suc}(x)$	=I
4.	$\mathbb{suc}(x) = Sx$	2,3 =E

(b) PA $\vdash \mathbb{zero}(x) = \emptyset$

(c) PA $\vdash \mathbb{idnt}^j_k(x_1 \dots x_j) = x_k$

(d) PA $\vdash \mathbb{plus}(x, y) = x + y$

(e) PA $\vdash \mathbb{times}(x, y) = x \times y$

(1) above is a first application of T13.21, with $\mathbb{suc}(x)$ equivalent to the original formula. Arguments for (a) - (c) are very much the same and nearly trivial. Arguments for (d) and (e) are by IN. Here is the case for (d) as an example.

1.	$gplus(x) = idnt^1_1(x)$	def from plus, T13.21
2.	$gplus(x) = x$	1 with T13.34c
3.	$plus(x, \emptyset) = gplus(x)$	T13.33a
4.	$plus(x, \emptyset) = x$	3,2 =E
5.	$x + \emptyset = x$	T6.39
6.	$plus(x, \emptyset) = x + \emptyset$	4,5 =E
7.	$plus(x, j) = x + j$	A (g →I)
8.	$plus(x, Sj) = hplus(x, j, plus(x, j))$	T13.33b
9.	$hplus(x, j, u) = suc(idnt^3_3(x, j, u))$	def from plus, T13.21
10.	$hplus(x, j, u) = Su$	9 with T13.34a,c
11.	$hplus(x, j, plus(x, j)) = S\,plus(x, j)$	10 ∀E
12.	$plus(x, Sj) = S\,plus(x, j)$	8,11 =E
13.	$plus(x, Sj) = S(x + j)$	12,7 =E
14.	$S(x + j) = x + Sj$	T6.40
15.	$plus(x, Sj) = x + Sj$	13,14 =E
16.	$[plus(x, j) = x + j] \rightarrow [plus(x, Sj) = x + Sj]$	7-15 →I
17.	$\forall y([plus(x, y) = x + y] \rightarrow [plus(x, Sy) = x + Sy])$	16 ∀I
18.	$plus(x, y) = x + y$	6,17 IN

Again, we simply write down the expressions on (1) and (9) with T13.21; then on (3) and (8) T13.33 makes the conditions for $plus(x, y)$ work like the ones for $x + y$ — so that with zero and inductive cases, the equivalence results by IN.

So this theorem establishes the equivalences we expect for the defined symbols suc, $zero$, $idnt$, $plus$ and $times$. It is important that $+$, $\times$ and the like are primitive symbols of $\mathscr{L}_{NT}$ where $plus$ and $times$ are defined according to our induction from the corresponding recursive functions. Having shown that the functions are equivalent, however, we may manipulate the one with all the results we have achieved for the other.

Some additional results will be facilitated by a couple of auxiliary definitions. $pred(y)$, $sg(y)$ and $csg(y)$ are defined directly, without appeal to recursive functions — but still behave as we expect.

Def[*pred*] PA $\vdash pred(y) = y \dot{-} \overline{1}$

Since this is a composition of functions, immediate by T13.17.

Def[*sg*] PA $\vdash v = sg(y) \leftrightarrow (y = \emptyset \wedge v = \emptyset) \vee (y > \emptyset \wedge v = S\emptyset)$

(i) PA $\vdash \exists v[(y = \emptyset \wedge v = \emptyset) \vee (y > \emptyset \wedge v = \overline{1})]$

(ii) PA $\vdash \forall u \forall v([(y = \emptyset \wedge u = \emptyset) \vee (y > \emptyset \wedge u = \overline{1})] \rightarrow [(y = \emptyset \wedge v = \emptyset) \vee (y > \emptyset \wedge v = \overline{1})])$

$Def[csg]$ PA $\vdash v = csg(y) \leftrightarrow (y = \emptyset \wedge v = \overline{1}) \vee (y > \emptyset \wedge v = \emptyset)$

(i) PA $\vdash \exists v[(y = \emptyset \wedge v = \overline{1}) \vee (y > \emptyset \wedge v = \emptyset)]$

(ii) PA $\vdash \forall u \forall v([(y = \emptyset \wedge u = \overline{1}) \vee (y > \emptyset \wedge u = \emptyset)] \rightarrow [(y = \emptyset \wedge v = \overline{1}) \vee (y > \emptyset \wedge v = \emptyset)])$

And some basic results on these notions,

T13.35. The following result in PA.

(a) PA $\vdash pred(\emptyset) = \emptyset$

(b) PA $\vdash pred(\overline{1}) = \emptyset$

(c) PA $\vdash y > \emptyset \rightarrow Spred(y) = y$

(d) PA $\vdash pred(Sy) = y$

(e) PA $\vdash y = \emptyset \leftrightarrow sg(y) = \emptyset$

(f) PA $\vdash y > \emptyset \leftrightarrow sg(y) = \overline{1}$

(g) PA $\vdash y = \emptyset \leftrightarrow csg(y) = \overline{1}$

(h) PA $\vdash y > \emptyset \leftrightarrow csg(y) = \emptyset$

(a) - (d) recover from $\dot{-}$ some basic results for *pred*; (b) is a simple particular result. (e) and (f) extract from the definition basic information for the behavior of *sg*; and (g) and (h) for *csg*.

And given these notions in PA, we can build on them for another set of equivalents.

*T13.36. The following result in PA.

(a) PA $\vdash \mathbb{p}red(y) = pred(y)$

*(b) PA $\vdash \mathbb{s}ubc(x, y) = x \dot{-} y$

(c) PA $\vdash \mathbb{a}bsval(x - y) = (x \dot{-} y) + (y \dot{-} x)$

(d) PA $\vdash \mathbb{s}g(y) = sg(y)$

(e) PA $\vdash \mathbb{c}sg(y) = csg(y)$

*(f) PA $\vdash \mathbb{E}q(x, y) \leftrightarrow x = y$

(g) PA $\vdash \mathbb{L}eq(x, y) \leftrightarrow x \leq y$

(h) PA $\vdash \mathbb{L}ess(x, y) \leftrightarrow x < y$

*(i) PA $\vdash \mathbb{N}eg(\mathbb{P}(\vec{x})) \leftrightarrow \sim\mathbb{P}(\vec{x})$

(j) PA $\vdash \mathbb{D}sj(\mathbb{P}(\vec{x}), \mathbb{Q}(\vec{y})) \leftrightarrow \mathbb{P}(\vec{x}) \vee \mathbb{Q}(\vec{y})$

Hints. (b): This works in the usual way up to the point in the show stage where you get $\mathbb{s}ubc(x, Sj) = \mathbb{p}red(x \dot{-} j)$; then it will take some work to show $x \dot{-} Sj = \mathbb{p}red(x \dot{-} j)$; for this begin with $x \leq j \vee x > j$ by T13.13p; the first case is straightforward; for the second, you will be able to show, $S(x \dot{-} Sj) = S\mathbb{p}red(x \dot{-} j)$ and apply T6.38. (f): For this relation, you have $\mathbb{E}q(x, y) \leftrightarrow \mathbb{s}g(\mathbb{a}bsval(x - y)) = \emptyset$ from the def EQ and T13.32; this gives $\mathbb{E}q(x, y) \leftrightarrow [(x \dot{-} y) + (y \dot{-} x)] = \emptyset$; now for $\leftrightarrow$I, the case from $x = y$ is easy; from $\mathbb{E}q(x, y)$, you have $x \geq y \vee x < y$ from T13.13p; the cases are not hard and similar (since $x < y$ gives $y \geq x$). (i): This is straightforward with $\mathbb{P}(\vec{x}) \leftrightarrow \mathbb{c}h_{\mathsf{P}}(\vec{x}) = \emptyset$ and $\mathbb{N}eg(\mathbb{P}(\vec{x})) \leftrightarrow \mathbb{c}sg(\mathbb{c}h_{\mathsf{P}}(\vec{x})) = \emptyset$ from NEG with T13.32.

So this theorem delivers the equivalences we expect for $\mathbb{p}red$, $\mathbb{s}ubc$, $\mathbb{a}bsval$, $\mathbb{s}g$, $\mathbb{c}sg$, $\mathbb{E}q$, $\mathbb{L}eq$, $\mathbb{L}ess$, $\mathbb{N}eg$, and $\mathbb{D}sj$. Given this, we will typically move without comment from some PA $\vdash \mathbb{D}sj(\mathbb{A}, \mathbb{B})$ given from T13.32 to PA $\vdash \mathbb{A} \vee \mathbb{B}$. And similarly in other cases.

We pause to remark on a on a simple consequence for characteristic functions. Recall from (CF) that a characteristic function is (officially) of the sort $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}}))$ so that,

T13.37. For any recursive characteristic function $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$, PA $\vdash \mathbb{c}h_{\mathsf{R}}(\vec{x}) = \emptyset \vee \mathbb{c}h_{\mathsf{R}}(\vec{x}) = \overline{1}$.

From (CF), $\mathsf{ch}_{\mathsf{R}}(\vec{\mathsf{x}})$ is of the sort $\mathsf{sg}(\mathsf{p}(\vec{\mathsf{x}}))$; so with T13.21, PA $\vdash \mathbb{c}h_{\mathsf{R}}(\vec{x}) = \mathbb{s}g(\mathbb{p}(\vec{x}))$. The result is nearly immediate with PA $\vdash \mathbb{p}(\vec{x}) = \emptyset \vee \mathbb{p}(\vec{x}) > \emptyset$ and results for $\mathbb{s}g$.

It is worth observing that this theorem, which depends on results for functions through T13.36d, results before any *use* of T13.32. There is therefore no problem about appeal to T13.37 in the demonstration of T13.32.

Now reasoning for the bounded quantifiers, bounded minimization and a couple relations built on them.

*T13.38. The following result in PA.

*(a) PA $\vdash (\boldsymbol{\exists} y \le z)\mathbb{P}(\vec{x}, z, y) \leftrightarrow (\exists y \le z)\mathbb{P}(\vec{x}, z, y)$

(b) PA $\vdash (\boldsymbol{\exists} y < z)\mathbb{P}(\vec{x}, z, y) \leftrightarrow (\exists y < z)\mathbb{P}(\vec{x}, z, y)$

(c) PA $\vdash (\boldsymbol{\forall} y \le z)\mathbb{P}(\vec{x}, z, y) \leftrightarrow (\forall y \le z)\mathbb{P}(\vec{x}, z, y)$

(d) PA $\vdash (\boldsymbol{\forall} y < z)\mathbb{P}(\vec{x}, z, y) \leftrightarrow (\forall y < z)\mathbb{P}(\vec{x}, z, y)$

*(e) PA $\vdash (\boldsymbol{\mu} y \le z)\mathbb{P}(\vec{x}, z, y) \leftrightarrow (\mu y \le z)\mathbb{P}(\vec{x}, z, y)$

(f) PA $\vdash \mathbb{F}ctr(m, n) \leftrightarrow m|n$

*(g) PA $\vdash \mathbb{P}rime(n) \leftrightarrow Pr(n)$

Hints. (a): Recall from chapter 12 that $\mathsf{s}(\vec{\mathsf{x}}, \mathsf{z}) = (\boldsymbol{\exists}\mathsf{y} \le \mathsf{z})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$ is defined by means of a $\mathsf{R}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{n})$ corresponding to $(\boldsymbol{\exists}\mathsf{y} \le \mathsf{n})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$; the main argument is to show by IN that PA $\vdash \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, n) = \emptyset \leftrightarrow (\exists y \le n)\mathbb{P}(\vec{x}, z, y)$. You have $\mathbb{P}(\vec{x}, z, y) \leftrightarrow \mathbb{ch}_{\mathsf{P}}(\vec{x}, z, y) = \emptyset$ from T13.32. For the zero case, you have $\mathbb{ch}_{\mathsf{R}}(\vec{x}, z, \emptyset) = \mathbb{gch}_{\mathsf{R}}(\vec{x}, z)$ from T13.33a, and $\mathbb{gch}_{\mathsf{R}}(\vec{x}, z) = \mathbb{ch}_{\mathsf{P}}(\vec{x}, z, \emptyset)$ from the definition with T13.21; for the main reasoning, you have $\mathbb{ch}_{\mathsf{R}}(\vec{x}, z, Sj) = \mathbb{hch}_{\mathsf{R}}(\vec{x}, z, j, \mathbb{ch}_{\mathsf{R}}(x, z, j))$ from T13.33b, and $\mathbb{hch}_{\mathsf{R}}(\vec{x}, z, j, u) = \mathbb{times}[u, \mathbb{ch}_{\mathsf{P}}(\vec{x}, z, \mathbb{suc}(j))]$ from the definition with T13.21; once you have finished the induction, it is a simple matter of applying $\mathbb{ch}_{\mathsf{S}}(\vec{x}, z) = \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, z)$ from the definition and T13.21, and where where $\mathbb{S}(\vec{x}, z)$ just abbreviates $(\boldsymbol{\exists} y \le z)\mathbb{P}(\vec{x}, z, y)$, applying $\mathbb{S}(\vec{x}, z) \leftrightarrow \mathbb{ch}_{\mathsf{S}}(\vec{x}, z) = \emptyset$ from T13.32 to get $(\boldsymbol{\exists} y \le z)\mathbb{P}(\vec{x}, z, y) \leftrightarrow (\exists y \le z)\mathbb{P}(\vec{x}, z, y)$. (f) and (g): Give previous results, these have nearly matching definitions except that the recursive side includes a bounded quantifier — so that you have to work to show the bound obtains for one direction of the biconditional.

The argument for T13.38e is particularly involved. Recall from chapter 12 that $\mathsf{m}(\vec{\mathsf{x}}, \mathsf{z}) = (\boldsymbol{\mu}\mathsf{y} \le \mathsf{z})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$ is defined by means of $\mathsf{R}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{n})$ as above and a $\mathsf{q}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{n})$ corresponding to $(\boldsymbol{\mu}\mathsf{y} \le \mathsf{n})\mathsf{P}(\vec{\mathsf{x}}, \mathsf{z}, \mathsf{y})$. The main reasoning is by IN to show $\mathbb{q}(\vec{x}, z, n) = (\mu y \le n)\mathbb{P}(\vec{x}, z, y)$; here are the main outlines of that part.

1.	$\mathbb{q}(\vec{x}, z, \emptyset) = (\mu y \leq \emptyset)\mathbb{P}(\vec{x}, z, y)$	[a]
2.	$\mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j) = \emptyset \vee \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j) = \overline{1}$	T13.37
3.	$\mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j) = \emptyset \leftrightarrow (\exists y \leq j)\mathbb{P}(x, z, y)$	from T13.38a
4.	$\mathbb{q}(\vec{x}, z, Sj) = \mathbb{hq}(\vec{x}, z, j, \mathbb{q}(\vec{x}, z, j))$	T13.33b
5.	$\mathbb{hq}(\vec{x}, z, j, u) = \mathbb{plus}(u, \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j))$	def from least, T13.21
6.	$\mathbb{hq}(\vec{x}, z, j, u) = u + \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j)$	5 T13.34d
7.	$\mathbb{hq}(\vec{x}, z, j, \mathbb{q}(\vec{x}, z, j)) = \mathbb{q}(\vec{x}, z, j) + \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j)$	6 $\forall$E
8.	$\mathbb{q}(\vec{x}, z, Sj) = \mathbb{q}(\vec{x}, z, j) + \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j)$	4,7 =E
9.	$\mathbb{q}(\vec{x}, z, j) = (\mu y \leq j)\mathbb{P}(\vec{x}, z, y)$	A ($g \to$I)
10.	$a = \mathbb{q}(\vec{x}, z, j)$	abv
11.	$b = \mathbb{q}(\vec{x}, z, Sj)$	abv
12.	$b = a + \mathbb{ch}_{\mathsf{R}}(\vec{x}, z, j)$	8,10,11 =E
13.	$a = (\mu y \leq j)\mathbb{P}(\vec{x}, z, y)$	9,10 =E
14.	$a = \mu y[y = j \vee \mathbb{P}(\vec{x}, z, y)]$	13 def
15.	$(\forall w < a)[w \neq j \wedge \sim\mathbb{P}(\vec{x}, z, w)]$	14 T13.19c
16.	$a = j \vee \mathbb{P}(\vec{x}, z, a)$	14 T13.19b
17.	$a = j$	A (g 16$\vee$E)
18.	$\sim\mathbb{P}(\vec{x}, z, j) \vee \mathbb{P}(\vec{x}, z, j)$	T3.1
19.	$\sim\mathbb{P}(\vec{x}, z, j)$	A (g 18$\vee$E)
20.	$[b = Sj \vee \mathbb{P}(\vec{x}, z, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, z, w))$	[b]
21.	$\mathbb{P}(\vec{x}, z, j)$	A (g18$\vee$E)
22.	$[b = Sj \vee \mathbb{P}(\vec{x}, z, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, z, w))$	[c]
23.	$[b = Sj \vee \mathbb{P}(\vec{x}, z, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, z, w))$	18,19-20,21-22 $\vee$E
24.	$\mathbb{P}(\vec{x}, z, a)$	A (g 16$\vee$E)
25.	$[b = Sj \vee \mathbb{P}(\vec{x}, z, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, z, w))$	[d]
26.	$[b = Sj \vee \mathbb{P}(\vec{x}, z, b)] \wedge (\forall w < b)(w \neq Sj \wedge \sim\mathbb{P}(\vec{x}, z, w))$	16,17-23,24-25 $\vee$E
27.	$b = \mu y[y = Sj \vee \mathbb{P}(\vec{x}, z, j)]$	24 def μ
28.	$b = (\mu y \leq Sj)\mathbb{P}(\vec{x}, z, y)$	27 def
29.	$\mathbb{q}(\vec{x}, z, Sj) = (\mu y \leq Sj)\mathbb{P}(\vec{x}, z, y)$	28 abv
30.	$[\mathbb{q}(\vec{x}, z, j) = (\mu y \leq j)\mathbb{P}(\vec{x}, z, y)] \to [\mathbb{q}(\vec{x}, z, Sj) = (\mu y \leq Sj)\mathbb{P}(\vec{x}, z, y)]$	9-29 $\to$I
31.	$\forall n([\mathbb{q}(\vec{x}, z, n) = (\mu y \leq n)\mathbb{P}(\vec{x}, z, y)] \to [\mathbb{q}(\vec{x}, z, Sn) = (\mu y \leq Sn)\mathbb{P}(\vec{x}, z, y)])$	30 $\forall$I
32.	$\mathbb{q}(\vec{x}, z, n) = (\mu y \leq n)\mathbb{P}(\vec{x}, z, y)$	1,31 IN

Hints: The zero case (a) is straightforward with T13.20a; for (b) you will be able to show that $b = Sj$; for (c) and (d) you will be able to show $b = a$. And the final result is nearly automatic from this.

T13.38 delivers the equivalences we expect for the bounded quantifiers, bounded minimization, factor and prime.

At this stage, we have defined in PA functions and relations corresponding to the recursive functions and relations. And we have taken advantage of equivalences to functions and relations already defined. Thus we are in a position simply to write down the following.

T13.39. The following are theorems of PA:

(a) PA $\vdash \mathbb{M}p(m,n,o) \leftrightarrow \mathbb{c}nd(n,o) = m$

(b) PA $\vdash \mathbb{I}con(m,n,o) \leftrightarrow \mathbb{M}p(m,n,o) \vee (m = n \wedge \mathbb{G}en(n,o))$

(c) PA $\vdash \mathbb{P}rft(m,n) \leftrightarrow \mathbb{e}xp(m, \mathbb{l}en(m) \dot{-} \overline{1}) = n \wedge m > \overline{1} \wedge (\forall k < \mathbb{l}en(m))[\mathbb{A}xiomt(\mathbb{e}xp(m,k)) \vee (\exists i < k)(\exists j < k)\mathbb{I}con(\mathbb{e}xp(m,i), \mathbb{e}xp(m,j), \mathbb{e}xp(m,k))]$

These follow directly from our results with their definitions MP, ICON and PRFQ. The definition with T13.32 gives us, say, PA $\vdash \mathbb{M}p(m,n,o) \leftrightarrow \mathbb{E}q(\mathbb{c}nd(n,o), m)$; then with T13.36f, we arrive at (a). And similarly in other cases.

Where $\mathbb{M}p$, $\mathbb{c}nd$ and the like are defined relative to corresponding recursive functions, it is important that the *operators* in expressions above are the ordinary operators of $\mathscr{L}_{\text{NT}}$. Thus we shall be able to manipulate them in the usual ways. We shall find these results useful for the following!

E13.22. Produce derivations to show T13.33a and T13.34e. Hard core: show the remaining cases from T13.34.

E13.23. Show (i) of the condition for *Def*[*pred*] and then T13.35c. Hard core: Show each of the conditions for *Def*[*pred*], *Def*[*sg*] and *Def*[*csg*] and all of the results in T13.35.

*E13.24. Show a, g and j from T13.36. Hard core: Demonstrate each of the results in T13.36.

*E13.25. Show T13.38a. Hard core: show T13.37 along with each of the results in T13.38.

Further results. Where T13.39 is interesting and important, for the second condition we shall require some further results especially involving functions from chapter 12 up to concatenation and well-formed formulas. Thus we begin with some results for exponentiation, factorial and the like upon which concatenation depends. In this case, we shall be acquiring results, not by demonstrating equivalence to expressions already defined (since there are no such expressions already defined), but directly for symbols defined from the recursive funcions.

*T13.40. The following are theorems of PA.

(a) (i) PA $\vdash m^{\emptyset} = \overline{1}$

(ii) PA $\vdash m^{Sn} = m^n \times m$

(b) PA $\vdash m^{\overline{1}} = m$

(c) PA $\vdash a > \emptyset \rightarrow \emptyset^a = \emptyset$

(d) PA $\vdash m^a \times m^b = m^{a+b}$

(e) PA $\vdash m \geq n \rightarrow m^a \geq n^a$

(f) PA $\vdash pred(m^b)|m^{a+b}$

(g) PA $\vdash (a > \emptyset \wedge m > \overline{1}) \rightarrow pred(m^{a+b}) \nmid m^b$

(h) PA $\vdash m > \emptyset \rightarrow m^a > \emptyset$

(i) PA $\vdash (m > \emptyset \wedge a \geq b) \rightarrow m^a \geq m^b$

(j) PA $\vdash (m > \overline{1} \wedge a > b) \rightarrow m^a > m^b$

(k) PA $\vdash a > \emptyset \rightarrow m^a \geq m$

*(l) PA $\vdash m > \overline{1} \rightarrow a < m^a$

Hints: (a) is from the the definition of power and prior results. (d) uses IN on the value of b and (e) uses IN on a. (f) is straightforward with cases for $m^b = \emptyset$ and $m^b > \emptyset$. (h), (i), (j) and (l) are by IN.

(a) gives the recursive conditions from which the rest follow. Then (b) - (l) are basic results that should be accessible from ordinary arithmetic.

*T13.41. The following are theorems of PA.

(a) (i) PA $\vdash fact(\emptyset) = \overline{1}$

(ii) PA $\vdash fact(Sn) = fact(n) \times Sn$

(b) PA $\vdash fact(\overline{1}) = \overline{1}$

(c) PA $\vdash fact(n) > \emptyset$

(d) PA $\vdash (\forall y < n)y|fact(n)$

*(e) $(\exists y \leq fact(n) + \overline{1})[n < y \wedge Pr(y)]$

Hints: (a) is from the definition of fact and prior results. (c) and (d) are straightforward by IN. Reasoning for (e) is like (G2) in the arithmetic for Gödel numbering reference once you realize that all the primes less than n are included in $fact(n)$.

These are some basic results for factorial. Again (a) gives the recursive conditions from which the rest follow. (b) is a simple particular fact; and the result from (c) is obvious. (d) is a consequence of the way the factorial includes all the numbers less than it. We will be able to take advantage of (e) immediately below.

*T13.42. The following are theorems of PA.

(a) (i) PA $\vdash \mathbb{pi}(\emptyset) = \overline{2}$

(ii) PA $\vdash \mathbb{pi}(Sn) = (\mu y \leq fact(\mathbb{pi}(n)) + \overline{1})[\mathbb{pi}(n) < y \wedge Pr(y)]$

(b) $(\exists y \leq fact(\mathbb{pi}(n)) + \overline{1})[\mathbb{pi}(n) < y \wedge Pr(y)]$

(c) PA $\vdash \mathbb{pi}(Sn) = \mu y[\mathbb{pi}(n) < y \wedge Pr(y)]$

(d) PA $\vdash \mathbb{pi}(n) < \mathbb{pi}(Sn) \wedge Pr(\mathbb{pi}(Sn))$

(e) PA $\vdash (\forall w < \mathbb{pi}(Sn)){\sim}[\mathbb{pi}(n) < w \wedge Pr(w)]$

(f) PA $\vdash Pr(\mathbb{pi}(n))$

(g) PA $\vdash \mathbb{pi}(n) > \overline{1}$

(h) PA $\vdash \mathbb{pi}(n)^a > \emptyset$

(i) PA $\vdash Spred(\mathbb{pi}(n)^a) = \mathbb{pi}(n)^a$

(j) PA $\vdash (\forall m < n)\mathbb{pi}(m) < \mathbb{pi}(n)$

(k) PA $\vdash (\forall m \leq n)Sm < \mathbb{pi}(n)$

*(l) PA $\vdash \forall y[Pr(y) \rightarrow \exists j\, \mathbb{pi}(j) = y]$

*(m) PA $\vdash m \neq n \rightarrow pred(\mathbb{pi}(m)) \nmid \mathbb{pi}(n)^a$

(n) PA $\vdash m \neq n \rightarrow pred(\mathbb{pi}(m)^{Sb}) \nmid \mathbb{pi}(n)^a$

*(o) PA $\vdash [m \neq n \wedge pred(\mathbb{pi}(m)^b)|(s \times \mathbb{pi}(n)^a)] \rightarrow pred(\mathbb{pi}(m)^b)|s$

Hints: (a) is from definition pi and prior results. (b) is from T13.41e; (c) applies T13.20.b; and then (d) and (e) are by T13.19(b) and (c). (f), (j) and (k) are simple inductions. (l) is by using IN to show $(\forall y \leq \mathbb{pi}(i))[Pr(y) \rightarrow \exists j\, \mathbb{pi}(j) = y]$; the result then follows easily with (k). Under the assumption for $\rightarrow$I, (m) is by IN on a. For (n) you will be able to show that if $pred(\mathbb{pi}(m)^{Sb})|\mathbb{pi}(n)^a$ then $pred(\mathbb{pi}(m))|\mathbb{pi}(n)^a$ and use (m). For (o) under the assumption for $\rightarrow$I you will be able to show $i \leq b \rightarrow pred(\mathbb{pi}(m)^i)|s$ by induction on i; the result then follows easily with $b \leq b$.

These are some basic results from prime sequences. (a) gives the basic recursive conditions. (b) is an existential result; then (c) extracts the successor condition from bounded to unbounded minimization; this allows application of the definition in (d) and (e). This is a first instance of a pattern we shall see repeatedly: Given a bounded condition $\mathbb{a} = (\mu x \leq t)\mathcal{P}(x)$ of the sort that arises from a recursive definition with T13.21, we show there exists some $\mathcal{P}(x)$ less than or equal to the bound; this allows application of T13.20.b to "extract" the bounded to an unbounded minimization, and then T13.19 to obtain $\mathcal{P}(\mathbb{a})$; this forms the basis for further results. (f) - (i) are some simple consequences of the fact that $\mathbb{pi}(n)$ is prime. Then the primes are ordered (j). And (k) each prime is greater than the successor of its index. (l) for any prime y, there is some j such that $\mathbb{pi}(j) = y$. And (m) - (o) echo results for factor except combined with primes and exponentiation.

In order to manipulate $\mathbb{exp}$, it will be convenient to introduce a function *ex*, that finds the least exponent x such that $\mathbb{pi}(i)^x$ does *not* divide Sn.

Def[*ex*] $ex(n,i) = \mu x[pred(\mathbb{pi}(i)^x) \nmid Sn]$

(i) PA $\vdash \exists x[pred(\mathbb{pi}(i)^x) \nmid Sn]$

1.	$\mathbb{pi}(i) > \overline{1}$	T13.42g
2.	$Sn < \mathbb{pi}(i)^{Sn}$	1 T13.40l
3.	$Spred(\mathbb{pi}(i)^{Sn}) = \mathbb{pi}(i)^{Sn}$	T13.42i
4.	$Sn < Spred(\mathbb{pi}(i)^{Sn})$	2,3 =E
5.	$n < pred(\mathbb{pi}(i)^{Sn})$	4 T13.13j
6.	$pred(\mathbb{pi}(i)^{Sn}) \nmid Sn$	5 T13.24i
7.	$\exists x[pred(\mathbb{pi}(i)^x) \nmid Sn]$	6 $\exists$I

*T13.43. The following are theorems of PA.

(a) PA $\vdash \mathbb{exp}(n,i) = (\mu x \leq n)[pred(\mathbb{pi}(i)^x)|n \wedge pred(\mathbb{pi}(i)^{x+\overline{1}}) \nmid n]$

(b) PA $\vdash \mathbb{exp}(\emptyset, i) = \emptyset$

*(c) PA $\vdash \mathbb{exp}(Sn, i) = \mu x[pred(\mathbb{pi}(i)^x)|Sn \wedge pred(\mathbb{pi}(i)^{x+\bar{1}}) \nmid Sn]$

(d) PA $\vdash pred(\mathbb{pi}(i)^{\mathbb{exp}(Sn,i)})|Sn \wedge pred(\mathbb{pi}(i)^{\mathbb{exp}(Sn,i)+\bar{1}}) \nmid Sn$

(e) PA $\vdash (\forall w < \mathbb{exp}(Sn, i)) \sim [pred(\mathbb{pi}(i)^w)|Sn \wedge pred(\mathbb{pi}(i)^{w+\bar{1}}) \nmid Sn]$

(f) PA $\vdash [pred(\mathbb{pi}(i)^a|Sn \wedge pred(\mathbb{pi}(i)^{a+\bar{1}} \nmid Sn] \rightarrow \mathbb{exp}(Sn, i) = a$

(g) PA $\vdash \mathbb{exp}(m, j) \leq m$

(h) PA $\vdash j \geq n \rightarrow \mathbb{exp}(Sn, j) = \emptyset$

(i) PA $\vdash \mathbb{exp}(\mathbb{pi}(i)^p, i) = p$

(j) PA $\vdash pred(\mathbb{pi}(i))|Sm \leftrightarrow \mathbb{exp}(Sm, i) \geq \bar{1}$

*(k) PA $\vdash \exists q[\mathbb{pi}(i)^{\mathbb{exp}(Sn,i)} \times q = Sn \wedge pred(\mathbb{pi}(i)) \nmid q \wedge \forall y(y \neq i \rightarrow \mathbb{exp}(q, y) = \mathbb{exp}(Sn, y))]$

*(l) PA $\vdash (m > \emptyset \wedge n > \emptyset) \rightarrow \mathbb{exp}(m \times n, i) = \mathbb{exp}(m, i) + \mathbb{exp}(n, i)$

Hints: (a) is from definition exp and prior results. (c) is by PA $\vdash (\exists x \leq Sn)[pred(\mathbb{pi}(i)^x)|Sn \wedge pred(\mathbb{pi}(i)^{x+\bar{1}}) \nmid Sn]$ and then T13.20b; $ex(n, i) = \emptyset \vee ex(n, i) > \emptyset$; in the latter case, the trick is to generalize on the number prior to $ex(n, i)$. (f) is by showing that $a = \mu x[pred(\mathbb{pi}(i)^x)|Sn \wedge pred(\mathbb{pi}(i)^{x+\bar{1}}) \nmid Sn]$. (k): from $pred(\mathbb{pi}(i)^{\mathbb{exp}(Sn,i)})|Sn$ there is a j such that $\mathbb{pi}(i)^{\mathbb{exp}(Sn,i)} \times j = Sn$; the hard part is to show $k \neq i \rightarrow \mathbb{exp}(j, k) = \mathbb{exp}(Sn, k)$ — for this, it will be helpful to establish that j is a successor. (l): under the assumption for $\rightarrow$I establish that m and n are successors; toward an application of T13.43f it will be easy to establish that $pred(\mathbb{pi}(i)^{\mathbb{exp}(m,i)+\mathbb{exp}(n,i)})|m \times n$; for the other, it will be helpful to begin with a couple applications of T13.43k.

(a) is from the definition. (b) is the standard result with bound $\emptyset$. (c) extracts the successor case from the bounded to an unbounded minimization; this allows application of the definition in (d) and (e). From (f) the reasoning goes the other way around: not only does the condition apply to the exponent, but if the condition applies to some a, then a is the exponent. Then (g) the exponent of some prime in the factorization of m cannot be greater than m; and (h) a prime whose index is greater than or equal to n does not divide into Sn. (i) makes an obvious connection for the exponent of a

prime, and (j) between exponent and factor. According (k) once you divide Sn by $\mathbb{pi}(i)$ $\mathbb{exp}(Sn, i)$ times you are left with a q such that $\mathbb{pi}(i)$ does not divide into it any more, and such that the exponents of all the other primes remain the same as in Sm. From (l) the i^{th} exponent of a product sums the i^{th} exponents of its factors.

*T13.44. The following are theorems of PA.

(a) PA $\vdash \mathbb{len}(n) = (\mu y \leq n)(\forall z \leq n)[z \geq y \rightarrow \mathbb{exp}(n, z) = \emptyset]$

(b) PA $\vdash \mathbb{len}(\emptyset) = \emptyset$

(c) PA $\vdash \mathbb{len}(Sn) = \mu y(\forall z \leq Sn)[z \geq y \rightarrow \mathbb{exp}(Sn, z) = \emptyset]$

(d) PA $\vdash (\forall z \leq Sn)[z \geq \mathbb{len}(Sn) \rightarrow \mathbb{exp}(Sn, z) = \emptyset]$

(e) PA $\vdash (\forall w < \mathbb{len}(Sn)){\sim}(\forall z \leq Sn)[z \geq w \rightarrow \mathbb{exp}(Sn, z) = \emptyset]$

(f) PA $\vdash \mathbb{len}(\overline{1}) = \emptyset$

(g) PA $\vdash \mathbb{len}(m) > \emptyset \rightarrow m > \overline{1}$

*(h) PA $\vdash \mathbb{exp}(m, i) > \emptyset \rightarrow \mathbb{len}(m) > i$

(i) PA $\vdash m > \overline{1} \rightarrow \mathbb{len}(m) > \emptyset$

*(j) PA $\vdash p > \emptyset \rightarrow \mathbb{len}(\mathbb{pi}(i)^p) = Si$

(k) PA $\vdash (\forall z \geq \mathbb{len}(n))\mathbb{exp}(n, z) = \emptyset$

*(l) PA $\vdash \mathbb{len}(n) = Sl \rightarrow \mathbb{exp}(n, l) \geq 1$

Hints: (a) is from definition length and prior results. (c) follows with T13.43h and existentially generalizing on Sn itself. (f) is by application of (c). Under the assumption for $\rightarrow$I, (h) divides into cases for $m = \emptyset$ and $m > \emptyset$; for the latter, suppose $\mathbb{len}(m) \not> i$; then you will be able to make use of (d). (i) is straightforward with T13.25d and ultimately (h) above. For (j), begin with $\mathbb{len}(\mathbb{pi}(i)^p) < Si \vee \mathbb{len}(\mathbb{pi}(i)^p) = Si \vee \mathbb{len}(\mathbb{pi}(i)^p) > Si$ by T13.13o; the first is easily eliminated with T13.44h; then, supposing $\mathbb{len}(\mathbb{pi}(i)^p) > Si$, you will be able to obtain a contradiction using T13.44e. (k): under the assumption $a \geq \mathbb{len}(n)$ for ($\forall$I), either $n = \emptyset$ or $n > \emptyset$; the first case is easy; for the second, there is some m such that $n = Sm$; your main reasoning will be to show $\mathbb{exp}(Sm, a) = \emptyset$. (l): under the assumption for $\rightarrow$I, the case when $n = \emptyset$ is impossible; so there is some m such that $n = Sm$; with this, suppose $\mathbb{exp}(Sm, l) \not\geq \overline{1}$; then you you will be able to show, contrary to your assumption that $\mathbb{len}(Sm) = l$.

Again (a) is from the definition and (b) gives the standard result for bound $\emptyset$. (c) extracts the successor case from bounded to unbounded minimization; (d) and (e) then apply the definition. (f) is a simple particular result; and then (g) is an immediate consequence of (b) and (f). From (h) if an exponent of some prime in the factorization of m is greater than zero, that prime is involved in the factorization of m; (i) gives the biconditional from (g); (j) gives the length for a prime to any power; and from (k) primes $\geq$ the length of n must all have exponent $\emptyset$. Length is set up so that it finds the first prime such that it and all the ones after have exponent zero; so (l) the prime prior to the length has exponent $\geq \emptyset$.

To manipulate concatenation, it will be helpful to introduce a couple of auxiliary notions. First, $exc(m,n,i)$ which (indirectly) takes the value of the i^{th} exponent in the concatenation of m and n.

$$\begin{aligned}\text{PA} \vdash\ & exc(m,n,i) = (\mu y \leq \mathbb{exp}(m,i) + \mathbb{exp}(n, i \dot{-} \mathbb{len}(m)))\\ & ([i < \mathbb{len}(m) \land y = \mathbb{exp}(m,i)] \lor [i \geq \mathbb{len}(m) \land y = \mathbb{exp}(n, i \dot{-} \mathbb{len}(m))])\end{aligned}$$

Since the definition is by bounded minimization, no condition is required. The idea is simply to set y to one or the other of $\mathbb{exp}(m,i)$ or $\mathbb{exp}(n, i \dot{-} \mathbb{len}(m))$ so that y takes the value of the i^{th} exponent in the concatenation of m and n. Then $val^*(m,n,i)$ is defined by recursion as follows.

$$\text{PA} \vdash val^*(m,n,\emptyset) = \overline{1}$$
$$\text{PA} \vdash val^*(m,n,Sy) = val^*(m,n,y) \times \mathbb{pi}(y)^{exc(m,n,y)}$$

So $val^*(m,n,i)$ returns the product of the first i primes in the factorization of the concatenation of m and n. It will also be helpful to have a notion $val(n,i)$ defined directly from exponents of its argument,

$$\text{PA} \vdash val(n,\emptyset) = \overline{1}$$
$$\text{PA} \vdash val(n,Sy) = val(n,y) \times \mathbb{pi}(y)^{exp(n,y)}$$

Again, $val(n,i)$ returns the product of the first i primes in the factorization of n. Say $m * n$ is the defined correlate to $\mathsf{m} \star \mathsf{n}$ and let $l = \mathbb{len}(m) + \mathbb{len}(n)$.

*T13.45. The following are theorems of PA.

(a) (i) PA $\vdash m * n = (\mu x \leq B_{m,n})[(\forall i < \mathbb{len}(m))\{\mathbb{exp}(x,i) = \mathbb{exp}(m,i)\} \land (\forall i < \mathbb{len}(n))\{\mathbb{exp}(x, i + \mathbb{len}(m)) = \mathbb{exp}(n,i)\}]$

(ii) PA $\vdash B_{m,n} = [\mathbb{pi}(l)^{m+n}]^l$

(b) PA $\vdash exc(m,n,i) = \mu y([i < \mathbb{len}(m) \land y = \mathbb{exp}(m,i)] \lor [i \geq \mathbb{len}(m) \land y = \mathbb{exp}(n, i \dot{-} \mathbb{len}(m))])$

(c) PA $\vdash i < \mathbb{len}(m) \to exc(m,n,i) = \mathbb{exp}(m,i)$

(d) PA $\vdash i \geq \mathbb{len}(m) \to exc(m,n,i) = \mathbb{exp}(n, i \dot{-} \mathbb{len}(m))$

(e) PA $\vdash val^*(m,n,i) > \emptyset$

*(f) PA $\vdash (\forall i \geq a) pred(\mathbb{pi}(i)) \nmid val^*(m,n,a)$

*(g) PA $\vdash (\forall j < i)\mathbb{exp}(val^*(m,n,i), j) = exc(m,n,j)$

*(h) PA $\vdash (\forall i < \mathbb{len}(m))[\mathbb{exp}(val^*(m,n,l), i) = \mathbb{exp}(m,i)] \wedge$
$(\forall i < \mathbb{len}(n))[\mathbb{exp}(val^*(m,n,l), i + \mathbb{len}(m)) = \mathbb{exp}(n,i)]$

*(i) PA $\vdash i \leq l \to [\mathbb{pi}(l)^{m+n}]^i \geq val^*(m,n,i)$

(j) PA $\vdash m{*}n = \mu x[(\forall i < \mathbb{len}(m))\{\mathbb{exp}(x,i) = \mathbb{exp}(m,i)\} \wedge (\forall i < \mathbb{len}(n))\{\mathbb{exp}(x, i + \mathbb{len}(m)) = \mathbb{exp}(n,i)\}]$

(k) PA $\vdash (\forall i < \mathbb{len}(m))\{\mathbb{exp}(m * n, i) = \mathbb{exp}(m,i)\} \wedge (\forall i < \mathbb{len}(n))\{\mathbb{exp}(m * n, i + \mathbb{len}(m)) = \mathbb{exp}(n,i)\}$

(l) PA $\vdash (\forall w < m * n){\sim}[(\forall i < \mathbb{len}(m))\{\mathbb{exp}(w,i) = \mathbb{exp}(m,i)\} \wedge (\forall i < \mathbb{len}(n))\{\mathbb{exp}(w, i + \mathbb{len}(m)) = \mathbb{exp}(n,i)\}]$

*(m) PA $\vdash \mathbb{len}(m * n) \geq l$

*(n) PA $\vdash \mathbb{len}(m * n) = l$

(o) PA $\vdash (\forall i < k)[\mathbb{exp}(a,i) = \mathbb{exp}(b,i) \to val(a,k) = val(b,k)]$

(p) PA $\vdash val(Sn, \mathbb{len}(Sn)) = Sn$

(q) PA $\vdash (\forall y < \mathbb{len}(n))[val(m * n, y + \mathbb{len}(m)) \geq val(m, \mathbb{len}(m))]$

corollary: PA $\vdash m * n \geq m$

(r) PA $\vdash (\forall y < \mathbb{len}(n))[val(m * n, y + \mathbb{len}(m)) \geq val(n, y)]$

corollary: PA $\vdash m * n \geq n$

Hints: (a) is from the definition concatenation with prior results. (f) is by IN on a. (g) is by IN on i; in the show under $(\forall j < i)\mathbb{exp}(val^*(m,n,i), j) = exc(m,n,j)$ and $a < Si$ you will have separate cases for $a < i$ and $a = i$. (h) is straightforward with applications of (g), (c) and (d). (i) is an induction on i; in the show, the main task is to obtain $exc(m,n,i) \leq m + n$; the

result then follows with previously established inequalities. (m) divides into cases for $\mathbb{len}(n) = \emptyset$ and $\mathbb{len}(n) > \emptyset$; and within the first, again, cases for $\mathbb{len}(m) = \emptyset$ and $\mathbb{len}(m) > \emptyset$. For (n) show $\mathbb{len}(m * n) \leq l$ and apply (m); for the main argument (which will be long!) assume $\mathbb{len}(m * n) \not\leq l$; then you will be able to apply T13.43k and show that the q so obtained contradicts T13.45l.

A great deal of work goes into obtaining (i) as a basis for (j); the idea is the same as behind the intuitive account of the bound from chapter 12: $\mathbb{pi}(l)^{m+n}$ is greater than every term in the factorization of $m * n$; so by a "simple" induction, for each $i \leq l$, $[\mathbb{pi}(l)^{m+n}]^i$ remains greater than $val^*(m, n, i)$; and $val^*(m, n, l)$ is therefore both under the bound and satisfies the condition for $m * n$ — so that the existential condition is satisfied, and we may extract the bounded to an unbounded minimization. Once this is acccomplished, we are most of the way home.

(a) is from the definition. Then (b) extracts *exc* from the bounded to unbounded minimization; and (c) and (d) apply the definition. (e) is obvious. (f) results because $val^*(m, n, a)$ is a product of primes prior to $\mathbb{pi}(a)$ so that greater primes do not divide it. Then (g) the exponents in in val^* are like the exponents in *exc*. This gives us (h) that the exponents in val^* are like the exponents in m and n. But val^* is constructed so that an induction enables a natural comparison between $B_{m,n}$ and $val^*(m, n, i)$ so that (i), up to any i, $[\mathbb{pi}(l)^{m+n}]^i \geq val^*(m, n, i)$. This enables us to extract $m * n$ from bounded to unbounded minimization (j) and apply the definition (k) and (l). (m) and (n) establish that the length of $m * n$ sums the lengths of m and n. Then (o) and (p) are some results for *val* which are applied in (q) and (r) for relative values of $m * n$.

Our last results in this section concern $\mathbb{F}ormseq$ and $\mathbb{W}ff$. Let,

$$\begin{array}{lcl}
A(s,x) & = & \mathbb{A}tomic(\mathbb{exp}(s,x)) \\
B(s,x) & = & (\exists j < x)[\mathbb{exp}(s,x) = \overline{\ulcorner\sim\urcorner} * \mathbb{exp}(s,j)] \\
C(s,x) & = & (\exists i < x)(\exists j < x)[\mathbb{exp}(s,x) = \mathbb{exp}(s,i) * \overline{\ulcorner\rightarrow\urcorner} * \mathbb{exp}(s,j)] \\
D(s,x) & = & (\exists i < x)(\exists j < p)[\mathbb{var}(j) \wedge \mathbb{exp}(s,x) = \overline{\ulcorner\forall\urcorner} * j * \exp \mathbb{exp}(s,i)]
\end{array}$$

*T13.46. The following are theorems of PA.

(a) PA $\vdash \mathbb{F}ormseq(m,n) \leftrightarrow \mathbb{exp}(m, \mathbb{len}(m) \mathbin{\dot{-}} \overline{1}) = n \wedge m > \overline{1} \wedge (\forall k < \mathbb{len}(m))[A(m,k) \vee B(m,k) \vee C(m,k) \vee D(m,k)]$

(b) (i) PA $\vdash \mathbb{W}ff(n) \leftrightarrow (\exists x \leq B_n)\mathbb{F}ormseq(x,n)$

(ii) PA $\vdash B_n = [\mathbb{pi}(\mathbb{len}(n))^n]^{\mathbb{len}(n)}$

(c) PA $\vdash \mathbb{F}ormseq(m,n) \to (\forall k < \mathbb{l}en(m))\mathbb{e}xp(m,k) > \overline{1}$

(d) PA $\vdash \mathbb{W}\!f\!f(p) \to p > \overline{1}$

(e) PA $\vdash (\forall i \leq n)\mathcal{P}(i) \leftrightarrow (\forall i \leq n)\mathcal{P}(n \dot{-} i)$

(f) PA $\vdash (\forall i \leq a)(\exists j : i < j \leq a)\mathbb{e}xp(n,i) \leq \mathbb{e}xp(n,j) \to (\forall i \leq a)\mathbb{e}xp(n,i) \leq \mathbb{e}xp(n,a)$

*(g) PA $\vdash \exists x \mathbb{F}ormseq(x,p) \to \exists x[\mathbb{F}ormseq(x,p) \wedge (\forall i < \mathbb{l}en(x) \dot{-} \overline{1})(\exists j : i < j \leq \mathbb{l}en(x) \dot{-} \overline{1})\mathbb{e}xp(x,i) \leq \mathbb{e}xp(x,j)]$

(h) PA $\vdash \exists x \mathbb{F}ormseq(x,p) \to \mathbb{W}\!f\!f(p)$

(i) PA $\vdash \mathbb{W}\!f\!f(\mathbb{n}eg(p)) \leftrightarrow \mathbb{W}\!f\!f(p)$

(j) PA $\vdash \mathbb{W}\!f\!f(\mathbb{c}nd(p,q)) \leftrightarrow \mathbb{W}\!f\!f(p) \wedge \mathbb{W}\!f\!f(q)$

(k) PA $\vdash \mathbb{P}rvt(p) \to \mathbb{W}\!f\!f(p)$

[These results (and some from the last) need cleanup. Esp. need that the length of the sequence is less than that of p.] From the definition, $\mathbb{F}ormseq(m,n)$ does not immediately yield $\mathbb{W}\!f\!f(n)$ insofar as m might not be under B_n. The general idea for these theorems is that, given a formula sequence, there is a *least* formula sequence adequate to yield $\mathbb{W}\!f\!f(n)$. Thus we move from the existence of a formula sequence through (g) to a formula sequence of a particular sort, and so to (h). Other results follow in a natural way.

The argument for (g) is particularly long and involved. Here is the basic idea. Let,

$$\begin{array}{lcl}
B^*(s,x) & = & (\exists j < x)[j \neq a \wedge \mathbb{e}xp(s,x) = \overline{\ulcorner \sim \urcorner} * \mathbb{e}xp(s,j)] \\
C^*(s,x) & = & (\exists i < x)(\exists j < x)[i \neq a \wedge j \neq a \wedge \mathbb{e}xp(s,x) = \mathbb{e}xp(s,i) * \overline{\ulcorner \to \urcorner} * \mathbb{e}xp(s,j)] \\
D^*(s,x) & = & (\exists i < x)(\exists j < p)[i \neq a \wedge \mathbb{v}ar(j) \wedge \mathbb{e}xp(s,x) = \overline{\ulcorner \forall \urcorner} * j * \exp \mathbb{e}xp(s,i)]
\end{array}$$

Then (a) under the assumption that there is a formula sequence for p, there is a *least* formula sequence m for p; (b) under the assumption that the consequent is false, some a is such that its exponent is greater than that of every subsequent member of m; (c) by T13.43k there is a q whose exponents are like m except without the a^{th} prime — this q is not itself a formula sequence; but $r = q \times \mathbb{p}i(a)^{\mathbb{e}xp(m,Sa)}$, must number a formula sequence for p less than m (which will be impossible); (d) the exponents of r are like the exponents of m except that $\mathbb{e}xp(r,a) = \mathbb{e}xp(m,Sa)$; then (e) the length of r is the same as the length of m so that $\mathbb{e}xp(r, \mathbb{l}en(r) \dot{-} \overline{1}) = p$; for the final part, you will be able to show (f), $(\forall k < \mathbb{l}en(m))[A(m,k) \vee B^*(m,k) \vee$

$C^*(m,k) \vee D^*(m,k)]$ (here is where you get a conflict from the special nature of a); with this, under the assumption that $v < \mathbb{len}(r)$ either $v \neq a$ or $v = a$; with the first (g) it is straightforward to get $A(r,v) \vee B(r,v) \vee C(r,v) \vee D(r,v)$; and (h) with the second, you can instantiate k to Sa for the same result; with this, $\mathbb{F}ormseq(r, p)$ and you have your contradiction. In order to make progress, it will be important at different stages to show that key variables p, m, q, r are > 1.

Theorems to carry forward from 13.4.1

Together with the results from T13.39, the following of the theorems that we have achieved in this part have application for the sections that follow.

T13.44h PA $\vdash \mathbb{exp}(m,i) > \emptyset \to \mathbb{len}(m) > i$

T13.45k PA $\vdash (\forall i < \mathbb{len}(m))\{\mathbb{exp}(m*n,i) = \mathbb{exp}(m,i)\} \wedge (\forall i < \mathbb{len}(n))\{\mathbb{exp}(m*n, i + \mathbb{len}(m)) = \mathbb{exp}(n,i)\}$

T13.45n PA $\vdash \mathbb{len}(m*n) = \mathbb{len}(m) + \mathbb{len}(n)$

T13.46d PA $\vdash \mathbb{Wff}(p) \to p > \overline{1}$

T13.46i PA $\vdash \mathbb{Wff}(\mathbb{neg}(p)) \leftrightarrow \mathbb{Wff}(p)$

T13.46j PA $\vdash \mathbb{Wff}(\mathbb{cnd}(p,q)) \leftrightarrow \mathbb{Wff}(p) \wedge \mathbb{Wff}(q)$

T13.46k PA $\vdash \mathbb{Prvt}(p) \to \mathbb{Wff}(p)$

*E13.26. Show (d) and (i) from T13.40. Hard core: show each of the results from T13.40.

*E13.27. Show (d) and (e) from T13.41. Hard core: show each of the results from T13.41.

*E13.28. Show (j) and (k) from T13.42. Hard core: show each of the results from T13.42.

*E13.29. Show (c) and (f) from T13.43. Hard core: show each of the results from T13.43.

*E13.30. Show (f) and (k) from T13.44. Hard core: show each of the results from T13.44.

*E13.31. Show (b), (c), (j) and (m) from T13.45. Hard core: show each of the results from T13.45.

*E13.32. Show ... from T13.46. Hard core: show each of the results from T13.46.

13.4.2 The result

After all our preparation, we are ready to turn to the second condition, that PA $\vdash \Box(\mathcal{P} \to \mathcal{Q}) \to (\Box\mathcal{P} \to \Box\mathcal{Q})$. Again, given both $\Box(\mathcal{P} \to \mathcal{Q})$ and $\Box\mathcal{P}$ the idea is that there are j and k such that $\textsc{prft}(\mathsf{j}, \ulcorner\mathcal{P} \to \mathcal{Q}\urcorner)$ and $\textsc{prft}(\mathsf{k}, \ulcorner\mathcal{P}\urcorner)$ so that $\mathsf{l} = \mathsf{j}\star\mathsf{k}\star 2^{\ulcorner\mathcal{Q}\urcorner}$ numbers a proof of $\mathcal{Q}$. As it turns out, it will be convenient to have it in a form with free variables, PA $\vdash Prvt(cnd(p,q)) \to (Prvt(p) \to Prvt(q))$; the second condition then follows as an immediate corollary.

Observe that we have on the table expressions of the sort, $+$, *Plus* and *plus* — where the first is a primitive symbol of $\mathcal{L}_{\text{NT}}$, the second the original relation to capture the recursive function plus, and the last a function symbol defined from the recursive function. In view of demonstrated equivalences, we will tend to slide between them without notice. So, for example, given that $\langle\langle 2,2\rangle,4\rangle \in \mathsf{plus}$ by capture PA $\vdash Plus(\overline{2},\overline{2},\overline{4})$; and by demonstrated equivalences, PA $\vdash \overline{2} + \overline{2} = \overline{4}$ and PA $\vdash plus(\overline{2},\overline{2}) = \overline{4}$; and similarly in other cases. We require such a move at different stages in the following.

*T13.47. PA $\vdash Prvt(cnd(p,q)) \to (Prvt(p) \to Prvt(q))$. Corollary: PA $\vdash \Box(\mathcal{P} \to \mathcal{Q}) \to (\Box\mathcal{P} \to \Box\mathcal{Q})$.

1.	$\mathbb{P}rvt(\mathbb{c}nd(p,q))$	A (g $\rightarrow$I)
2.	$\mathbb{W}ff(\mathbb{c}nd(p,q))$	1 T13.46k
3.	$\mathbb{W}ff(p)$	2 T13.46j
4.	$\mathbb{W}ff(q)$	2 T13.46j
5.	$\mathbb{P}rvt(p)$	A (g $\rightarrow$I)
6.	$\mathbb{I}con(\mathbb{c}nd(p,q),p,q)$	T13.39a,b
7.	$\exists v\,\mathbb{P}rft(v,\mathbb{c}nd(p,q))$	1 abv
8.	$\exists v\,\mathbb{P}rft(v,p)$	5 abv
9.	$\mathbb{P}rft(j,\mathbb{c}nd(p,q))$	A (g 7$\exists$E)
10.	$\mathbb{P}rft(k,p)$	A (g 8$\exists$E)
11.	$l=(j*k)*2^q$	def
12.	$\mathbb{e}xp(j,\mathbb{l}en(j)\dot{-}\overline{1})=\mathbb{c}nd(p,q)$	9 T13.39c
13.	$\mathbb{e}xp(k,\mathbb{l}en(k)\dot{-}\overline{1})=p$	10 T13.39c
14.	$\mathbb{e}xp(l,\mathbb{l}en(j)+\mathbb{l}en(k))=q$	11 T13.45k,n
15[a]	$\mathbb{I}con[\mathbb{e}xp(j,\mathbb{l}en(j)\dot{-}\overline{1}),\mathbb{e}xp(k,\mathbb{l}en(k)\dot{-}\overline{1}),\mathbb{e}xp(l,\mathbb{l}en(j)+\mathbb{l}en(k))]$	6,12,13,14 =E
16.	$(\forall i<\mathbb{l}en(j))[\mathbb{e}xp(l,i)=\mathbb{e}xp(j,i)]$	11 T13.45k
17.	$(\forall i<\mathbb{l}en(k))[\mathbb{e}xp(l,\mathbb{l}en(j)+i)=\mathbb{e}xp(k,i)]$	11 T13.45k
18.	$\mathbb{e}xp(l,\mathbb{l}en(j)\dot{-}\overline{1})=\mathbb{e}xp(j,\mathbb{l}en(j)\dot{-}\overline{1})$	16 T13.44h ($\forall$E)
19.	$\mathbb{e}xp(l,\mathbb{l}en(j)+\mathbb{l}en(k)\dot{-}\overline{1})=\mathbb{e}xp(k,\mathbb{l}en(k)\dot{-}\overline{1})$	17 T13.44h ($\forall$E)
20[b]	$Icon[\mathbb{e}xp(l,\mathbb{l}en(j)\dot{-}\overline{1}),\mathbb{e}xp(l,\mathbb{l}en(j)+\mathbb{l}en(k)\dot{-}\overline{1}),\mathbb{e}xp(l,\mathbb{l}en(j)+\mathbb{l}en(k))]$	15,18,19 =E
21.	$(\forall i<\mathbb{l}en(j))[\mathbb{A}xiom(\mathbb{e}xp(l,i))\vee(\exists m<i)(\exists n<i)Icon(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,i))]$	9,16 T13.39c
22.	$(\forall i<\mathbb{l}en(k))[\mathbb{A}xiom(\mathbb{e}xp(l,\mathbb{l}en(j)+i))\vee$	
	$(\exists m<i)(\exists n<i)\mathbb{I}con(\mathbb{e}xp(l,\mathbb{l}en(j)+m),\mathbb{e}xp(l,\mathbb{l}en(j)+n),\mathbb{e}xp(l,\mathbb{l}en(j)+i))]$	10,17 T13.39c
23[c]	$(\forall i:\mathbb{l}en(j)\leq i<\mathbb{l}en(j)+\mathbb{l}en(k))[\mathbb{A}xiom(\mathbb{e}xp(l,i))\vee$	
	$(\exists m<i)(\exists n<i)\mathbb{I}con(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,i))]$	from 22
24.	$x<\mathbb{l}en(l)$	A (g ($\forall$I))
25.	$x<\mathbb{l}en(j)\vee\mathbb{l}en(j)\leq x<\mathbb{l}en(j)+\mathbb{l}en(k)\vee x=\mathbb{l}en(j)+\mathbb{l}en(k)$	11,24 T13.45n
26.	$x<\mathbb{l}en(j)$	A (g 25$\vee$E)
27.	$\mathbb{A}xiom(\mathbb{e}xp(l,x))\vee(\exists m<x)(\exists n<x)\mathbb{I}con(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,x))$	21,26 ($\forall$E)
28.	$\mathbb{l}en(j)\leq x<\mathbb{l}en(j)+\mathbb{l}en(k)$	A (g 25$\vee$E)
29.	$\mathbb{A}xiom(\mathbb{e}xp(l,x))\vee(\exists m<x)(\exists n<x)\mathbb{I}con(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,x))$	23 ($\forall$E)
30.	$x=\mathbb{l}en(j)+\mathbb{l}en(k)$	A (g (25$\vee$E)
31.	$(\exists m<x)(\exists n<x)\mathbb{I}con(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,x))$	20,30
32.	$\mathbb{A}xiom(\mathbb{e}xp(l,x))\vee(\exists m<x)(\exists n<x)\mathbb{I}con(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,x))$	31 $\vee$I
33.	$\mathbb{A}xiom(\mathbb{e}xp(l,x))\vee(\exists m<x)(\exists n<x)\mathbb{I}con(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,x))$	25 $\vee$E
34[d]	$(\forall x<\mathbb{l}en(l))[\mathbb{A}xiom(\mathbb{e}xp(l,x))\vee(\exists m<x)(\exists n<x)\mathbb{I}con(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,x))]$	24-33 ($\forall$I)
35.	$q>\emptyset$	4 T13.46d
36.	$\mathbb{l}en(2^q)=\overline{1}$	35 T13.44j
37.	$\mathbb{l}en(l)\geq\overline{1}$	11,36 T13.45n
38.	$l>\overline{1}$	T13.44g
39.	$\mathbb{e}xp(l,\mathbb{l}en(l)\dot{-}\overline{1})=q$	14 T13.45n
40.	$\mathbb{e}xp(l,\mathbb{l}en(l)\dot{-}\overline{1})=q\wedge l>\overline{1}\wedge(\forall x<\mathbb{l}en(l))[\mathbb{A}xiom(\mathbb{e}xp(l,x))\vee$	
41	$(\exists m<x)(\exists n<x)Icon(\mathbb{e}xp(l,m),\mathbb{e}xp(l,n),\mathbb{e}xp(l,x))]$	37,38,39 $\wedge$I
42.	$\mathbb{P}rft(l,q)$	40 T13.39c
43.	$\mathbb{P}rvt(q)$	42 $\exists$I
44.	$\mathbb{P}rvt(q)$	8,10-43 $\exists$E
45.	$\mathbb{P}rvt(q)$	7,9-44 $\exists$E
46.	$\mathbb{P}rvt(p)\rightarrow\mathbb{P}rvt(q)$	5-45 $\rightarrow$I
47[e]	$\mathbb{P}rvt(\mathbb{c}nd(p,q))\rightarrow[\mathbb{P}rvt(p)\rightarrow\mathbb{P}rvt(q)]$	1-46 $\rightarrow$I

This derivation is long, and skips steps; but it should be enough for you to see how the argument works — and to fill in the details if you choose. First, at (a), under assumptions for $\rightarrow$I, there are derivations numbered j, k and a longer sequence numbered l. And the the last member of this longer sequence is an immediate consequence of last members from the derivations numbered j and k. At (b) the results from (12) are all applied to the sequence numbered l; so the last sentence in the longer sequence is an immediate consequence of its earlier members. At (c), the different fragments of the longer sequence have the character of a proof. And at (d), the whole sequence numbered l has the character of a proof. Finally, at (e) we observe that this longer sequence yields $\mathbb{P}rvt(q)$ and discharge the assumptions for the result that $\mathbb{P}rvt(cnd(p,q)) \rightarrow [\mathbb{P}rvt(p) \rightarrow \mathbb{P}rvt(q)]$ so that with T13.32 PA $\vdash Prvt(cnd(p,q)) \rightarrow (Prvt(p) \rightarrow Prvt(q))$.

But then we have $Prvt(cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner})) \rightarrow [Prvt(\overline{\ulcorner \mathcal{P} \urcorner}) \rightarrow Prvt(\overline{\ulcorner \mathcal{Q} \urcorner})]$ as an instance, and by capture, $Prvt(\overline{\ulcorner \mathcal{P} \rightarrow \mathcal{Q} \urcorner}) \rightarrow [Prvt(\overline{\ulcorner \mathcal{P} \urcorner}) \rightarrow Prvt(\overline{\ulcorner \mathcal{Q} \urcorner})]$ so that PA $\vdash \Box(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\Box\mathcal{P} \rightarrow \Box\mathcal{Q})$. Thus the second derivability condition is established.

*E13.33. As a start to a complete demonstration of T13.47, provide a demonstration through part (c) that does not skip any steps. You may find it helpful to divide your demonstration into separate parts for (a), (c) and then for lines (18), (19) and (20). Hard core: complete the entire derivation.

13.5 The Third Condition: $\Box\mathcal{P} \rightarrow \Box\Box\mathcal{P}$

To show the third condition, that PA $\vdash \Box\mathcal{P} \rightarrow \Box\Box\mathcal{P}$, it is sufficient to show PA $\vdash \mathcal{Q} \rightarrow \Box\mathcal{Q}$. For when $\mathcal{Q}$ is $\Box\mathcal{P}$, the result is immediate. Further, $\Box\mathcal{P}$ is $Prvt(\overline{\ulcorner \mathcal{P} \urcorner})$; but $\overline{\ulcorner \mathcal{P} \urcorner}$ is a numeral, and $Prvt$ is Σ_1; so $Prvt(\overline{\ulcorner \mathcal{P} \urcorner})$ is Σ_1. So it is sufficient to show that for any Σ_1 sentence $\mathcal{Q}$, PA $\vdash \mathcal{Q} \rightarrow \Box\mathcal{Q}$. We build gradually to this result. Observe that, insofar as we appeal to theorems from before (including D2) the results of this section remain dependent on work from previous sections.

13.5.1 More applications

Recall from chapter 12 that where $\mathsf{p} = \ulcorner \mathcal{P} \urcorner$, $\mathsf{v} = \ulcorner v \urcorner$, and $\mathsf{s} = \ulcorner s \urcorner$ there is a recursive $\mathsf{formsub(p,v,s)}$ which returns the Gödel number of $\mathcal{P}^{v}_{s}$. In addition, there is a relation $\mathsf{num(n)}$ that returns the Gödel number of the standard numeral for n. Let $\mathsf{gvar(n)} =_{\text{def}} 2^{23+2\mathsf{n}}$ be the Gödel number of variable x_n. Then

$\mathsf{formsub}(\mathsf{p}, \mathsf{gvar}(\mathsf{n}), \mathsf{num}(\mathsf{y}))$ is a function which returns the number of the formula that substitutes a numeral for the value (number) assigned to y into the place of x_n. So, for example, if y is assigned the value of 2, then $\mathsf{formsub}(\mathsf{p}, \mathsf{gvar}(\mathsf{n}), \mathsf{num}(\mathsf{y}))$ returns $\ulcorner \mathcal{P}^{x_n}_{\overline{2}} \urcorner$. So PA defines $formsub(p, gvar(n), num(y))$. Now,

— this section is work in progress —

T13.48. The following are theorems of PA.

(a) If PA $\vdash \mathbb{P}rvt(p)$ then PA $\vdash \mathbb{P}rvt(\overline{\ulcorner\forall\urcorner} * gvar(\overline{\mathsf{n}}) * p)$

(b) PA $\vdash \mathbb{W}ff(p) \to Prvt(cnd(\overline{\ulcorner\forall\urcorner} * gvar(\overline{\mathsf{n}}) * p, formsub(p, gvar(\overline{\mathsf{n}}), num(x))))$

Where substituted terms are numerals (so that restrictions are automatically met), effectively, (a) is like Gen* and (b) like A4.

T13.49. The following are theorems of PA. Suppose $x = x_i$ and $y = x_j$.

(a) If x is not free in $\mathcal{P}$, then PA $\vdash formsub(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner x \urcorner}, y) = \overline{\ulcorner \mathcal{P} \urcorner}$

(b) PA $\vdash formsub(formsub(p, gvar(\overline{\mathsf{m}}), num(x_m)), gvar(\overline{\mathsf{n}}), num(x_n)) =$

$$formsub(formsub(p, gvar(\overline{\mathsf{n}}), num(x_n)), gvar(\overline{\mathsf{m}}), num(x_m))$$

(c) PA $\vdash formsub(cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), gvar(\overline{\mathsf{i}}), num(x)) =$

$$cnd(formsub(\overline{\ulcorner \mathcal{P} \urcorner}, gvar(\overline{\mathsf{i}}), num(x)), formsub(\overline{\ulcorner \mathcal{Q} \urcorner}, gvar(\overline{\mathsf{i}}), num(x)))$$

(d) PA $\vdash formsub(\overline{\ulcorner \mathcal{P}^x_y \urcorner}, gvar(\overline{\mathsf{j}}), num(y)) =$

$$formsub[formsub(\overline{\ulcorner \mathcal{P} \urcorner}, gvar(\overline{\mathsf{i}}), num(y)), gvar(\overline{\mathsf{j}}), num(y)].$$

(e) PA $\vdash formsub(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, gvar(\overline{\mathsf{j}}), num(y)) =$

$$formsub[formsub(\overline{\ulcorner \mathcal{P} \urcorner}, gvar(\overline{\mathsf{i}}), num(Sy)), gvar(\overline{\mathsf{j}}), num(y)].$$

(a) is obvious. From (b) substituting numerals for x_m and then x_n is the same as substituting for x_n and then x_m. (c) substituting into a conditional is the same as the conditional with substitutions into the antecedent and consequent. (d) substituting for y in $\overline{\ulcorner \mathcal{P}^x_y \urcorner}$ is the same as substituting for both x and y in $\overline{\ulcorner \mathcal{P} \urcorner}$ (catching x-place and any original y-places too). And, similarly, (e) substituting for y in $\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}$ is the same as replacing x with the numeral for Sy and y with the numeral for y in $\overline{\ulcorner \mathcal{P} \urcorner}$.

Theorems to carry forward from 13.5.1

Together with the results from T13.39, the following of the theorems that we have achieved in this part have application for the sections that follow.

T13.48a If PA $\vdash \mathbb{P}rvt(p)$ then PA $\vdash \mathbb{P}rvt(\overline{\ulcorner\forall\urcorner} * gvar(\overline{\mathsf{n}}) * p)$

T13.48b PA $\vdash \mathbb{W}ff(p) \to Prvt(cnd(\overline{\ulcorner\forall\urcorner} * gvar(\overline{\mathsf{n}}) * p,\ formsub(p, gvar(\overline{\mathsf{n}}), num(x))))$

T13.49a If x is not free in $\mathcal{P}$, then PA $\vdash formsub(\overline{\ulcorner\mathcal{P}\urcorner}, \overline{\ulcorner x\urcorner}, y) = \overline{\ulcorner\mathcal{P}\urcorner}$

T13.49b PA $\vdash formsub(formsub(p, gvar(\overline{\mathsf{m}}), num(x_m)), gvar(\overline{\mathsf{n}}), num(x_n)) =$
$formsub(formsub(p, gvar(\overline{\mathsf{n}}), num(x_n)), gvar(\overline{\mathsf{m}}), num(x_m))$

T13.49c PA $\vdash formsub(cnd(\overline{\ulcorner\mathcal{P}\urcorner}, \overline{\ulcorner\mathcal{Q}\urcorner}), gvar(\overline{\mathsf{i}}), num(x)) =$
$cnd(formsub(\overline{\ulcorner\mathcal{P}\urcorner}, gvar(\overline{\mathsf{i}}), num(x)), formsub(\overline{\ulcorner\mathcal{Q}\urcorner}, gvar(\overline{\mathsf{i}}), num(x)))$

T13.49d PA $\vdash formsub(\overline{\ulcorner\mathcal{P}^x_y\urcorner}, gvar(\overline{\mathsf{j}}), num(y)) =$
$formsub[formsub(\overline{\ulcorner\mathcal{P}\urcorner}, gvar(\overline{\mathsf{i}}), num(y)), gvar(\overline{\mathsf{j}}), num(y)]$.

T13.49e PA $\vdash formsub(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, gvar(\overline{\mathsf{j}}), num(y)) =$
$formsub[formsub(\overline{\ulcorner\mathcal{P}\urcorner}, gvar(\overline{\mathsf{i}}), num(Sy)), gvar(\overline{\mathsf{j}}), num(y)]$.

13.5.2 Substitutions

Return to our function $\mathsf{formsub(p, gvar(n), num(y))}$ which returns the number of the formula that substitutes a numeral for the value assigned to y into the place of vaiable x_n, and to the corresponding $formsub(p, gvar(n), num(y))$. We now define a $sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})$ which substitutes numerals for all the variables free in $\mathcal{P}$. Where $\vec{x}$ is a (possibly empty) sequence $x_1 \ldots x_n$ including at least all the free variables in $\mathcal{P}$,

$$\text{PA} \vdash sub_0(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = \overline{\ulcorner\mathcal{P}\urcorner}$$

$$\text{PA} \vdash sub_{Si}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = formsub(sub_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), gvar(S\overline{\mathsf{i}}), num(x_{Si}))$$

And PA $\vdash sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = sub_n(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})$. Observe that $sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})$ has as free variables the variables free in $\mathcal{P}$ but, intuitively, returns the Gödel number of a sentence — the sentence which substitutes into places for free variables numerals for the values assigned to those variables.

With T13.49a and T13.49b, we can show that so long as $\vec{x}$ and $\vec{y}$ include all the free variables of $\mathcal{P}$, $sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y})$. Thus,

*T13.50. If $\vec{x}$ and $\vec{y}$ are the same except that $\vec{y}$ includes some variables not in $\vec{x}$ (and so not free in $\mathcal{P}$), then PA $\vdash sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y})$.

Hint: Where the variables of $\vec{x}$ are ordered $x_{1.0}, x_{2.0} \ldots x_{n.0}$ let the variables of $\vec{y}$ be of the sort, $x_{0.1} \ldots x_{0.a}; x_{1.0} \ldots x_{1.b}; \ldots x_{n.0} \ldots x_{n.c}$. So $S(n.m)$ is either $n.Sm$ or $Sn.0$. Then by a simple induction on the value of $n.m$ you will be able to show that $sub_{n.0}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = sub_{n.m}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y})$.

*T13.51. PA $\vdash sub(\overline{\ulcorner\mathcal{P}\urcorner}, x_0, \vec{x}, \vec{y}) = sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}, x_0, \vec{y})$.

Observe that for any $\vec{x} = x_1 \ldots x_n$ the value of $sub_{n+1}(\overline{\ulcorner\mathcal{P}\urcorner}, x_0, \vec{x}, \vec{y})$ and of $sub_{n+1}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}, x_0, \vec{y})$ does not depend on the variables in $\vec{y}$. So, as a minor simplification, it is enough to concentrate on showing PA $\vdash sub_{n+1}(\overline{\ulcorner\mathcal{P}\urcorner}, x_0, x_1 \ldots x_n) = sub_{n+1}(\overline{\ulcorner\mathcal{P}\urcorner}, x_1 \ldots x_n, x_0)$.

The argument is an induction on the value of n. The key to this is that $sub_{i+2}(\overline{\ulcorner\mathcal{P}\urcorner}, x_1 \ldots x_{i+1}, x_0) = \mathit{formsub}[\mathit{formsub}(sub_i(\overline{\ulcorner\mathcal{P}\urcorner}, x_1 \ldots x_i), \mathit{gvar}(\overline{\mathsf{i}+1}), \mathit{num}(x_{i+1})), \mathit{gvar}(\overline{0}), \mathit{num}(x_0)]$; then you will be able to apply T13.49b and the assumption.

This effectively gives the ability to sort variables from one order into another. Suppose the members of $\vec{x}$ are in the standard order. To convert $\vec{y}$ to $\vec{x}$, a straightforward approach is to switch members into the first position in the reverse of their order in $\vec{x}$ — so for n members, at stage i, the result is $x_{Sn-i} \ldots x_n, \vec{y}'$ where $\vec{y}'$ is like $\vec{y}$ less the members that preceed it. So for a vector with 6 members, at stage 0 we begin with some $sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y})$; then at stage three PA proves this is equivalent to $sub(\overline{\ulcorner\mathcal{P}\urcorner}, x_4, x_5, x_6, \vec{y}')$; and at stage 6 that it is equivalent to $sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})$. This is an induction, but simple enough, so left as an exercise.

Given that PA $\vdash sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{y})$ for vectors including all the free variables in $\mathcal{P}$, simply select a standard vector with just the free variables in $\mathcal{P}$ and all the variables in a standard order. Then introducing double brackets as a special notation,

$$\mathit{Prvt}[\![\mathcal{P}(\vec{x})]\!] =_{\text{def}} \mathit{Prvt}(sub(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$$

Where $\mathcal{P}$ has free variables $\vec{x}$, $\mathit{Prvt}(\overline{\ulcorner\mathcal{P}\urcorner})$ asserts the provability of the open formula $\mathcal{P}(\vec{x})$. But $\mathit{Prvt}[\![\mathcal{P}(\vec{x})]\!]$ itself has all the free variables of $\mathcal{P}$ and asserts the provability of whatever *sentences* have numerals for the variables free in $\mathcal{P}$: so, for example, $\forall x \mathit{Prvt}[\![\mathcal{P}(x)]\!]$ asserts the provability of $\mathcal{P}^x_{\emptyset}$, $\mathcal{P}^x_{S\emptyset}$, and so forth. When $\mathcal{P}$ is a sentence, there are no substitutions to be made, and $\mathit{Prvt}[\![\mathcal{P}]\!]$ is the same as

$\mathit{Prvt}(\overline{\ulcorner\mathcal{P}\urcorner})$. Thus we set out to show PA $\vdash \mathcal{P} \to \mathit{Prvt}[\![\mathcal{P}]\!]$ for Σ_1 formulas. When $\mathcal{P}$ is a sentence, this gives PA $\vdash \mathcal{P} \to \mathit{Prvt}(\overline{\ulcorner\mathcal{P}\urcorner})$, which is to be shown.

Finally we shall require also some short theorems in order to manipulate this new notion. Each is by a short induction. First analogs to D1 and D2.

T13.52. If PA $\vdash \mathcal{P}$, then PA $\vdash \mathit{Prvt}[\![\mathcal{P}]\!]$ — analog to D1

Suppose PA $\vdash \mathcal{P}$. By induction on the value of n, PA $\vdash \mathit{Prvt}(\mathit{sub}_n(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$; the case when $i = n$ gives the desired result.

Basis: $\mathit{sub}_0(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = \overline{\ulcorner\mathcal{P}\urcorner}$. Since PA $\vdash \mathcal{P}$, by D1, PA $\vdash \mathit{Prvt}(\overline{\ulcorner\mathcal{P}\urcorner})$; so PA $\vdash \mathit{Prvt}(\mathit{sub}_0(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$.

Assp: PA $\vdash \mathit{Prvt}(\mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$.

Show: PA $\vdash \mathit{Prvt}(\mathit{sub}_{Si}(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$. By assumption, PA $\vdash \mathit{Prvt}(\mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$; so by T13.48a, PA $\vdash \mathbb{P}\mathit{rvt}(\overline{\ulcorner\forall\urcorner} * \mathbb{g}\mathit{var}(\overline{S\mathsf{i}}) * \mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}))$; so with T13.46k, $\mathbb{W}\mathit{ff}(\mathbb{P}\mathit{rvt}(\overline{\ulcorner\forall\urcorner} * \mathbb{g}\mathit{var}(\overline{S\mathsf{i}}) * \mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})))$; so by T13.48b, PA $\vdash \mathit{Prvt}(\mathbb{c}\mathit{nd}(\overline{\ulcorner\forall\urcorner} * \mathbb{g}\mathit{var}(\overline{S\mathsf{i}}) * \mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), \mathit{formsub}(\mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), \mathbb{g}\mathit{var}(\overline{S\mathsf{i}}), \mathbb{n}\mathit{um}(x_{Si}))))$; so with D2, PA $\vdash \mathit{Prvt}(\overline{\ulcorner\forall\urcorner} * \mathbb{g}\mathit{var}(\overline{S\mathsf{i}}) * \mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})) \to \mathit{Prvt}(\mathit{formsub}(\mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), \mathbb{g}\mathit{var}(\overline{S\mathsf{i}}), \mathbb{n}\mathit{um}(x_{Si})))$; so by $\to$E, PA $\vdash \mathit{Prvt}(\mathit{formsub}(\mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}), \mathbb{g}\mathit{var}(\overline{S\mathsf{i}}), \mathbb{n}\mathit{um}(x_{Si})))$; so with the definition, PA $\vdash \mathit{Prvt}(\mathit{sub}_{Si}(\overline{\ulcorner\mathcal{P}\urcorner}), \vec{x})$.

Indct: For any n, PA $\vdash \mathit{Prvt}(\mathit{sub}_n(\overline{\ulcorner\mathcal{P}\urcorner}), \vec{x})$

So PA $\vdash \mathit{Prvt}(\mathit{sub}(\overline{\ulcorner\mathcal{P}\urcorner}), \vec{x})$ and PA $\vdash \mathit{Prvt}[\![\mathcal{P}]\!]$.

T13.53. PA $\vdash \mathit{Prvt}[\![\mathcal{P} \to \mathcal{Q}]\!] \to (\mathit{Prvt}[\![\mathcal{P}]\!] \to \mathit{Prvt}[\![\mathcal{Q}]\!])$ — analog to D2

By induction on n, PA $\vdash \mathit{Prvt}(\mathit{sub}_n(\overline{\ulcorner\mathcal{P} \to \mathcal{Q}\urcorner}, \vec{x})) \to (\mathit{Prvt}(\mathit{sub}_n(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})) \to \mathit{Prvt}(\mathit{sub}_n(\overline{\ulcorner\mathcal{Q}\urcorner}, \vec{x})))$.

Basis: PA $\vdash \mathit{sub}_0(\overline{\ulcorner\mathcal{P} \to \mathcal{Q}\urcorner}, \vec{x}) = \overline{\ulcorner\mathcal{P} \to \mathcal{Q}\urcorner}$; PA $\vdash \mathit{sub}_0(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x}) = \overline{\ulcorner\mathcal{P}\urcorner}$; and PA $\vdash \mathit{sub}_0(\overline{\ulcorner\mathcal{Q}\urcorner}, \vec{x})) = \overline{\ulcorner\mathcal{Q}\urcorner}$. By D2, PA $\vdash \mathit{Prvt}(\overline{\ulcorner\mathcal{P} \to \mathcal{Q}\urcorner}) \to (\mathit{Prvt}(\overline{\ulcorner\mathcal{P}\urcorner}) \to \mathit{Prvt}(\overline{\ulcorner\mathcal{Q}\urcorner}))$; so PA $\vdash \mathit{Prvt}(\mathit{sub}_0(\overline{\ulcorner\mathcal{P} \to \mathcal{Q}\urcorner}, \vec{x})) \to (\mathit{Prvt}(\mathit{sub}_0(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})) \to \mathit{Prvt}(\mathit{sub}_0(\overline{\ulcorner\mathcal{Q}\urcorner}, \vec{x})))$.

Assp: PA $\vdash \mathit{Prvt}(\mathit{sub}_i(\overline{\ulcorner\mathcal{P} \to \mathcal{Q}\urcorner}, \vec{x})) \to (\mathit{Prvt}(\mathit{sub}_i(\overline{\ulcorner\mathcal{P}\urcorner}, \vec{x})) \to \mathit{Prvt}(\mathit{sub}_i(\overline{\ulcorner\mathcal{Q}\urcorner}, \vec{x})))$.

Show: PA $\vdash Prvt(sub_{Si}(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x})) \to (Prvt(sub_{Si}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to Prvt(sub_{Si}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})))$.

Suppose PA $\vdash Prvt(sub_{Si}(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x}))$. By capture, PA $\vdash \overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner} = cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner})$; so PA $\vdash Prvt(sub_{Si}(cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}))$. But,

(i) PA $\vdash sub_{Si}(cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}) = formsub(sub_i(cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}), gvar(S\overline{\text{i}}), num(x_{Si}))$;

(ii) PA $\vdash sub_{Si}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}) = formsub(sub_i(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), gvar(S\overline{\text{i}}), num(x_{Si}))$;

(iii) PA $\vdash sub_{Si}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}) = formsub(sub_i(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}), gvar(S\overline{\text{i}}), num(x_{Si}))$.

From PA $\vdash Prvt(sub_{Si}(cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}))$ we have with (i), PA $\vdash Prvt(formsub(sub_i(cnd(\overline{\ulcorner \mathcal{P} \urcorner}, \overline{\ulcorner \mathcal{Q} \urcorner}), \vec{x}), gvar(S\overline{\text{i}}), num(x_{Si})))$; so with T13.49c, PA $\vdash Prvt(cnd(formsub(sub_i(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), gvar(S\overline{\text{i}}), num(x_{Si})), formsub(sub_i(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}), gvar(S\overline{\text{i}}), num(x_{Si}))))$; so by =E with (ii) and (iii), PA $\vdash Prvt(cnd(sub_{Si}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x}), sub_{Si}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})))$; so with D2 and MP, PA $\vdash Prvt(sub_{Si}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to Prvt(sub_{Si}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x}))$. So by DT we have, PA $\vdash Prvt(sub_{Si}(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x})) \to (Prvt(sub_{Si}(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to Prvt(sub_{Si}(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})))$.

Indct: For any n, PA $\vdash Prvt(sub_n(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x})) \to (Prvt(sub_n(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to Prvt(sub_n(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})))$.

So PA $\vdash Prvt(sub(\overline{\ulcorner \mathcal{P} \to \mathcal{Q} \urcorner}, \vec{x})) \to (Prvt(sub(\overline{\ulcorner \mathcal{P} \urcorner}, \vec{x})) \to Prvt(sub(\overline{\ulcorner \mathcal{Q} \urcorner}, \vec{x})))$. And PA $\vdash Prvt[\![\mathcal{P} \to \mathcal{Q}]\!] \to (Prvt[\![\mathcal{P}]\!] \to Prvt[\![\mathcal{Q}]\!])$.

T13.54. If t is one of $\emptyset$, y or Sy and t is free for x in $\mathcal{P}$, then PA $\vdash Prvt[\![\mathcal{P}^x_t]\!] \leftrightarrow Prvt[\![\mathcal{P}]\!]^x_t$.

Consider the case $t = Sy$ and take the variables in the order $x, y, \vec{z}$. Observe that $Prvt[\![\mathcal{P}^x_{Sy}]\!] = Prvt(sub(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, x, y, \vec{z}))$. And $Prvt[\![\mathcal{P}]\!]^x_{Sy} = Prvt[sub(\overline{\ulcorner \mathcal{P} \urcorner}, x, y, \vec{z})]^x_{Sy} = Prvt[sub(\overline{\ulcorner \mathcal{P} \urcorner}, x, y, \vec{z})^x_{Sy}]$. Thus it suffices to show PA $\vdash sub(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, x, y, \vec{z}) = sub(\overline{\ulcorner \mathcal{P} \urcorner}, x, y, \vec{z})^x_{Sy}$. By induction, PA $\vdash sub_n(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, x, y, \vec{z}) = sub_n(\overline{\ulcorner \mathcal{P} \urcorner}, x, y, \vec{z})^x_{Sy}$.

Basis: Since x is not free in $\mathcal{P}^x_{Sy}$, with T13.49a, PA $\vdash sub_1(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, x, y, \vec{z}) = formsub(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, gvar(\overline{\text{i}}), num(x)) = \overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}$. And with T13.49e, PA $\vdash sub_2(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, x, y, \vec{z}) = formsub(\overline{\ulcorner \mathcal{P}^x_{Sy} \urcorner}, gvar(\overline{\text{j}}), num(y)) = formsub[formsub(\overline{\ulcorner \mathcal{P} \urcorner}, gvar(\overline{\text{i}}), num(Sy)), gvar(\overline{\text{j}}), num(y)]$. But PA $\vdash sub_1(\overline{\ulcorner \mathcal{P} \urcorner}, x, y, \vec{z})^x_{Sy} = formsub(\overline{\ulcorner \mathcal{P} \urcorner}, gvar(\overline{\text{i}}), num(Sy))$. So

PA $\vdash sub_2(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z})^x_{Sy} = formsub[formsub(\overline{\ulcorner\mathcal{P}\urcorner}, gvar(\overline{\mathsf{i}}), num(Sy)),$ $gvar(\overline{\mathsf{j}}), num(y)]$. So PA $\vdash sub_2(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, x, y, \vec{z}) = sub_2(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z})^x_{Sy}$.

Assp: For $2 \leq i$, PA $\vdash sub_i(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, x, y, \vec{z}) = sub_i(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z})^x_{Sy}$.

Show: PA $\vdash sub_{Si}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, x, y, \vec{z}) = sub_{Si}(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z})^x_{Sy}$.

PA $\vdash sub_{Si}(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, x, y, \vec{z}) = formsub(sub_i(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, x, y, \vec{z}), gvar(S\overline{\mathsf{i}}),$ $num(x_{Si})) =$ (by assp) $formsub(sub_i(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z})^x_{Sy}, gvar(S\overline{\mathsf{i}}), num(x_{Si}))$ $= formsub(sub_i(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z}), gvar(S\overline{\mathsf{i}}), num(x_{Si}))^x_{Sy} = sub_{Si}(\overline{\ulcorner\mathcal{P}\urcorner},$ $x, y, \vec{z})^x_{Sy}$.

Indct: PA $\vdash sub_n(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, x, y, \vec{z}) = sub_n(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z})^x_{Sy}$

So PA $\vdash sub(\overline{\ulcorner\mathcal{P}^x_{Sy}\urcorner}, x, y, \vec{z}) = sub(\overline{\ulcorner\mathcal{P}\urcorner}, x, y, \vec{z})^x_{Sy}$. And PA $\vdash Prvt[\![\mathcal{P}^x_{Sy}]\!] \leftrightarrow$ $Prvt[\![\mathcal{P}]\!]^x_{Sy}$. Other cases are similar and left for homework.

*E13.34. Provide a demonstration for T13.50

*E13.35. (i) Provide a demonstration for T13.51. (ii) Then provide a demonstration for the sorting result that is "simple enough" and so left as an exercise.

E13.36. Complete the demonstration of T13.54 by completing the remaining cases.

13.5.3 Sigma star.

Now we introduce an alternate characterization of the Σ_1 formulas — one that will result in a sort of simplification. Our aim is to demonstrate a result for all the Σ_1 formulas. Given our minimal resources, the task will be simplified if we can give a minimal specification of the Σ_1 formulas themselves. Toward this end, we introduce a special class of formulas, the $\Sigma^\star$ formulas; show that every Σ_1 formula is a $\Sigma^\star$ formula; and demonstrate our result with respect to this special class. Say a $\Sigma^\star$ formula is defined as follows.

($\Sigma^\star$) For any variables x, y and z,

(a) $\emptyset = z$, $y = z$, $Sy = z$, $x + y = z$ and $x \times y = z$ are $\Sigma^\star$ formulas.

(s) If $\mathcal{P}$ and $\mathcal{Q}$ are $\Sigma^\star$ formulas, then so are $(\mathcal{P} \vee \mathcal{Q})$, and $(\mathcal{P} \wedge \mathcal{Q})$.

($\forall$) If $\mathcal{P}$ is a $\Sigma^\star$ formula, then so is $(\forall x \le y)\mathcal{P}$ where y does not occur in $\mathcal{P}$.

($\exists$) If $\mathcal{P}$ is a $\Sigma^\star$ formula, then so is $\exists x\mathcal{P}$.

(c) Nothing else is a $\Sigma^\star$ formula.

We aim to show that any Σ_1 formula is provably equivalent to a $\Sigma^\star$ formula. Then results which apply to all the $\Sigma^\star$ formulas immediately transfer to the Σ_1 formulas. We begin showing that there are $\Sigma^\star$ formulas equivalent to atomic equalities of the sort $t = x$. Then (depending on an extended notion of *normal* form and a result result according to which Δ_0 formulas always have equivalent normal forms) we show that there are $\Sigma^\star$ formulas equivalent to Δ_0 formulas. From this it is a short step to the result that there are $\Sigma^\star$ formulas equivalent to all the Σ_1 formulas. First, then, the result for atomic equalities,

T13.55. For any $\mathcal{P}$ of the form $t = x$, there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

By induction on the function symbols in t.

Basis: If t has no function symbols, then it is the constant $\emptyset$ or a variable y, so $\mathcal{P}$ is of the form, $\emptyset = x$ or $y = x$; but these are already $\Sigma^\star$ formulas. So let $\mathcal{P}^\star$ be the same as $\mathcal{P}$. Then PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

Assp: For any i, $0 \le i < k$, if t has i function symbols, there is a $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

Show: If t has k function symbols, there is a $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

If t has k function symbols, then it is of the form, Sr, $r + s$ or $r \times s$ for r and s with $< k$ function symbols.

(S) t is Sr, so that $\mathcal{P}$ is $Sr = x$. Set $\mathcal{P}^\star = \exists z[(r = z)^\star \wedge Sz = x]$; then by assumption, PA $\vdash r = z \leftrightarrow (r = z)^\star$. So reason as follows,

1.	$r = z \leftrightarrow (r = z)^\star$	assp
2.	$\quad Sr = x$	A (g $\leftrightarrow$I)
3.	$\quad r = r \wedge Sr = x$	from 2
4.	$\quad \exists z[r = z \wedge Sz = x]$	3 $\exists$I
5.	$\quad \exists z[(r = z)^\star \wedge Sz = x]$	1,4 with T9.9
6.	$\quad \exists z[(r = z)^\star \wedge Sz = x]$	A (g $\leftrightarrow$I)
7.	$\quad\quad (r = z)^\star \wedge Sz = x$	A (g 6$\exists$E)
8.	$\quad\quad r = z$	1,7 $\leftrightarrow$E
9.	$\quad\quad Sr = x$	from 7,8
10.	$\quad Sr = x$	6,7-9 $\exists$E
11.	$Sr = x \leftrightarrow \exists z[(r = z)^\star \wedge Sz = x]$	2-5,6-10 $\leftrightarrow$I

So PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

(+) $t = s + r$, so that $\mathcal{P}$ is $s + r = x$. Set $\mathcal{P}^\star = \exists u \exists v[(s = u)^\star \wedge (r = v)^\star \wedge u + v = x]$. Then PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

($\times$) Similarly.

Indct: For any $\mathcal{P}$ of the form $t = x$, there is a $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

Now generalize the definitions for a *normal* form from T8.1. Thus, in an extended sense, say a formula is in *normal* form iff its only operators are $\vee$, $\wedge$, $\sim$, or a bounded quantifier, and the only instances of $\sim$ are immediately prefixed to atomics (which may include inequalities). Again, generalizing from before, where $\mathcal{P}$ is a normal form, let $\mathcal{P}'$ be like $\mathcal{P}$ except that $\vee$ and $\wedge$, universal and existential quantifiers and, for an atomic $\mathcal{A}$, $\mathcal{A}$ and $\sim\mathcal{A}$ are interchanged. So, for example, $(\exists x \leq p)(x = p \vee x \not> p)' = (\forall x \leq p)(x \neq p \wedge x > p)$. Finaly, then, for any Δ_0 formula whose operators are $\sim$, $\rightarrow$ and the bounded quantifiers, for atomic $\mathcal{A}$, $\mathcal{A}^* = \mathcal{A}$; $[\sim\mathcal{P}]^* = [\mathcal{P}^*]'$; $(\mathcal{P} \rightarrow \mathcal{Q})^* = ([\mathcal{P}^*]' \vee \mathcal{Q}^*)$; $[(\exists x \leq t)\mathcal{P}]^* = (\exists x \leq t)\mathcal{P}^*$ and $[(\forall x \leq t)\mathcal{P}]^* = (\forall x \leq t)\mathcal{P}^*$ (and similarly for $(\exists x < t)$ and $(\forall x < t)$). Then as a simple extension to the result from E8.9,

T13.56. For any Δ_0 formula $\mathcal{P}$, there is a normal formula $\mathcal{P}^*$ such that $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^*$.

The demonstration is straightforward extension of the reasoning from E8.9.

We show our result as applied to these normal forms. Thus,

*T13.57. For any Δ_0 formula $\mathcal{P}$ there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

From T13.56, for any Δ_0 formula $\mathcal{P}$, there is a normal $\mathcal{P}^*$ such that $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^*$. Now by induction on the number of operators in $\mathcal{P}^*$, we show PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$.

Basis: If $\mathcal{P}^*$ has no operators, then it is an atomic of the sort $s = t$, $s \leq t$ or $s < t$.

(=) $\mathcal{P}^*$ is $s = t$. Set $\mathcal{P}^\star = \exists z[(s = z)^\star \wedge (t = z)^\star]$. By T13.55, PA $\vdash s = z \leftrightarrow (s = z)^\star$ and PA $\vdash t = z \leftrightarrow (t = z)^\star$; so PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$.

($\leq$) $\mathcal{P}^*$ is $s \leq t$, which is to say $\exists z(z + s = t)$. By the case immediately above, PA $\vdash (z+s = t) \leftrightarrow (z+s = t)^\star$. Set $\mathcal{P}^\star = \exists z(z+s = t)^\star$. Then PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$. And similarly for $<$.

Assp: For any i, $0 \leq i < k$, if a normal $\mathcal{P}^*$ has i operator symbols, then there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$.

Show: If a normal $\mathcal{P}^*$ has k operator symbols, then there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$.

If $\mathcal{P}^*$ has k operator symbols, then it is of the form $\sim\mathcal{A}$, $\mathcal{B} \wedge \mathcal{C}$, $\mathcal{B} \vee \mathcal{C}$, $(\exists x \leq t)\mathcal{B}$, $(\exists x < t)\mathcal{B}$, $(\forall x \leq t)\mathcal{B}$ or $(\forall x < t)\mathcal{B}$, where $\mathcal{A}$ is atomic and $\mathcal{B}$ and $\mathcal{C}$ are normal with $< k$ operator symbols.

($\sim$) $\mathcal{P}^*$ is $\sim\mathcal{A}$. (i) $\mathcal{P}^*$ is $s \neq t$. Set $\mathcal{P}^\star = (s < t)^\star \vee (t < s)^\star$; then by assumption, PA $\vdash s < t \leftrightarrow (s < t)^\star$ and PA $\vdash t < s \leftrightarrow (t < s)^\star$; and with T13.13o, PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$.

(ii) $\mathcal{P}^*$ is $s \not< t$; set $\mathcal{P}^\star = (t \leq s)^\star$; then by assumption, PA $\vdash t \leq s \leftrightarrow (t \leq s)^\star$; and with T13.13q, PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$. And similarly for $\mathcal{P}^\star = s \not\leq t$.

($\wedge$) $\mathcal{P}^*$ is $\mathcal{B} \wedge \mathcal{C}$. Set $\mathcal{P}^\star = \mathcal{B}^\star \wedge \mathcal{C}^\star$; since $\mathcal{B}$ and $\mathcal{C}$ are normal, by assumption PA $\vdash \mathcal{B} \leftrightarrow \mathcal{B}^\star$ and PA $\vdash \mathcal{C} \leftrightarrow \mathcal{C}^\star$; so PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$. And similarly for $\vee$.

($\forall$) $\mathcal{P}^*$ is $(\forall x \leq t)\mathcal{B}$. Set $\mathcal{P}^\star = \exists z[(t = z)^\star \wedge (\forall x \leq z)\mathcal{B}^\star]$; by T13.55 PA $\vdash t = z \leftrightarrow (t = z)^\star$ and by assumption, PA $\vdash \mathcal{B} \leftrightarrow \mathcal{B}^\star$ so PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$. And, by a related construction, similarly for $(\forall x < t)\mathcal{B}$.

($\exists$) $\mathcal{P}^*$ is $(\exists x \leq t)\mathcal{B}$. Set $\mathcal{P}^\star = \exists x[(x \leq t)^\star \wedge \mathcal{B}^\star]$; then by assumption PA $\vdash x \leq t \leftrightarrow (x \leq t)^\star$ and PA $\vdash \mathcal{B} \leftrightarrow \mathcal{B}^\star$; so PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$. And similarly for $(\exists x < t)\mathcal{B}$.

Indct: For any normal $\mathcal{P}^*$ there is a $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$.

So for any Δ_0 formula $\mathcal{P}$, there is a $\mathcal{P}^*$ such that $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^*$ and now PA $\vdash \mathcal{P}^* \leftrightarrow \mathcal{P}^\star$. So PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

Now it is immediate that for any Σ_1 formula $\mathcal{P}$ there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

T13.58. For any Σ_1 formula $\mathcal{P}$ there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

Consider any Σ_1 formula $\mathcal{P}$. This formula is of the form $\exists x_1 \ldots \exists x_n \mathcal{A}$ for Δ_0 formula $\mathcal{A}$. But by T13.57, there is an $\mathcal{A}^\star$ such that PA $\vdash \mathcal{A} \leftrightarrow \mathcal{A}^\star$. Let $\mathcal{P}^\star$ be $\exists x_1 \ldots \exists x_n \mathcal{A}^\star$. Then PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

E13.37. Povide a demonstration to show T13.56.

*E13.38. Fill in the parts of T13.55 and T13.57 that are left as "similarly" to to show that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

13.5.4 The result.

And now we can show PA $\vdash \mathcal{P} \rightarrow \mathit{Prvt}[\![\mathcal{P}]\!]$ by induction on the number of operators in a $\Sigma^\star$ formula $\mathcal{P}$. From this, by the previous theorem, we have that PA $\vdash \mathcal{P} \rightarrow \mathit{Prvt}[\![\mathcal{P}]\!]$ for any Σ_1 formula $\mathcal{P}$. And this is the result we need for D3. The argument is by induction on the number of operators in a $\Sigma^\star$ formula. And this is aided by the special rules for construction of such formulas.

Before we launch into the main argument, a word about substitution. From their original statement, the rules $\forall$I and =E result in formulas of the sort $\mathcal{P}^x_t$ or $\mathcal{P}^t/_s$. So from, say, $\forall$E applied to $\forall x \mathit{Prvt}[\![\mathcal{P}]\!]$ we get something of the sort $\mathit{Prvt}[\![\mathcal{P}]\!]^x_t$. But we need to be careful about what the substitution comes to. In the simplest case, $\mathit{Prvt}[\![\mathcal{P}(x)]\!]$ is of the sort $\mathit{Prvt}(\mathit{formsub}(\overline{\ulcorner \mathcal{P}(x) \urcorner}, \mathit{gvar}(\overline{\mathrm{i}}), \mathit{num}(x)))$, where there is a free x to be replaced by t; but this does not automatically convert to $\mathit{Prvt}[\![\mathcal{P}(t)]\!]$ insofar as that requires application of *formsub* to $\overline{\ulcorner \mathcal{P}(t) \urcorner}$. But we do have a theorem, T13.54 which tells us that in certain cases PA $\vdash \mathit{Prvt}[\![\mathcal{P}^x_t]\!] \leftrightarrow \mathit{Prvt}[\![\mathcal{P}]\!]^x_t$, so that the replacements can be moved across the bracket in the natural way. With this said, we turn to our theorem.

T13.59. For any $\Sigma^\star$ formula $\mathcal{P}$, PA $\vdash \mathcal{P} \rightarrow \mathit{Prvt}[\![\mathcal{P}]\!]$.

By induction on the number of operators in $\mathcal{P}$.

Basis: If a $\Sigma^\star$ $\mathcal{P}$ has no operator symbols, then it is an atomic of the sort $\emptyset = z$, $y = z$, $Sy = z$, $x + y = z$ or $x \times y = z$.

(S) Suppose $\mathcal{P}$ is $Sy = z$. Reason as follows,

1.	$Sy = Sy$	=I
2.	$Prvt\llbracket Sy = Sy\rrbracket$	1 T13.52
3.	$\quad Sy = z$	A (g →I)
4.	$\quad Prvt\llbracket (Sy = z)^{z}_{Sy}\rrbracket$	2 abv
5.	$\quad Prvt\llbracket Sy = z\rrbracket^{z}_{Sy}$	4 T13.54
6.	$\quad Prvt\llbracket Sy = z\rrbracket$	3,5 =E
7.	$Sy = z \rightarrow Prvt\llbracket Sy = z\rrbracket$	3-6 →I

Observe that T13.52 applies to theorems, and so not to formulas under the assumption for →I. Thus we take care to restrict its application to formulas against the main scope line. Also, at (5) we use T13.54 to move the substitution across the bracket. With this done, the substitution on line (4) applies only to the free z of $Prvt\llbracket Sy = z\rrbracket$ — that is, to the free z of $Prvt(sub(\overline{\ulcorner Sy = z\urcorner}, y, z)$; so that =E applies in a straightforward way to substitute a z back into that place. The argument is similar for $\emptyset = z$ and $y = z$.

(+) Suppose $\mathcal{P}$ is $x+y = z$. The proof in PA requires appeal to IN, with induction on the value of x in $\forall y\forall z(x + y = z \rightarrow Prvt\llbracket x + y = z\rrbracket)$. For the basis,

1.	$\emptyset + y = y$	T6.49
2.	$Prvt\llbracket \emptyset + y = y\rrbracket$	1 T13.52
3.	$\quad (x + y = z)^{x}_{\emptyset}$	A (g →I)
4.	$\quad \emptyset + y = z$	3 abv
5.	$\quad y = z$	1,4 =E
6.	$\quad Prvt\llbracket (\emptyset + y = z)^{z}_{y}\rrbracket$	2 abv
7.	$\quad Prvt\llbracket \emptyset + y = z\rrbracket^{z}_{y}$	6 T13.54
8.	$\quad Prvt\llbracket \emptyset + y = z\rrbracket$	6,5 =E
9.	$\quad Prvt\llbracket (x + y = z)^{x}_{\emptyset}\rrbracket$	8 abv
10.	$\quad Prvt\llbracket x + y = z\rrbracket^{x}_{\emptyset}$	9 T13.54
11.	$(x + y = z)^{x}_{\emptyset} \rightarrow Prvt\llbracket x + y = z\rrbracket^{x}_{\emptyset}$	3-10 →I
12.	$(x + y = z \rightarrow Prvt\llbracket x + y = z\rrbracket)^{x}_{\emptyset}$	11 abv
13.	$\forall y\forall z(x + y = z \rightarrow Prvt\llbracket x + y = z\rrbracket)^{x}_{\emptyset}$	12 ∀I

And the inductive stage,

14.	$x + Sy = z \leftrightarrow Sx + y = z$	T6.40,T6.51
15.	$Prvt[\![x + Sy = z \to Sx + y = z]\!]$	14 T13.52
16.	$\forall y \forall z(x + y = z \to Prvt[\![x + y = z]\!])$	A (g →I)
17.	$(x + y = z)^{x}_{Sx}$	A (g →I)
18.	$Sx + y = z$	17 abv
19.	$x + Sy = z$	14,18 ↔E
20.	$x + Sy = z \to Prvt[\![x + y = z]\!]^{y}_{Sy}$	16 ∀E
21.	$Prvt[\![x + y = z]\!]^{y}_{Sy}$	20,19 →E
22.	$Prvt[\![x + Sy = z]\!]$	21 T13.54
23.	$Prvt[\![x + Sy = z]\!] \to Prvt[\![Sx + y = z]\!]$	15 T13.53
24.	$Prvt[\![Sx + y = z]\!]$	23,22 →E
25.	$Prvt[\![x + y = z]\!]^{x}_{Sx}$	24 T13.54
26.	$(x + y = z)^{x}_{Sx} \to Prvt[\![x + y = z]\!]^{x}_{Sx}$	17-25 →I
27.	$(x + y = z \to Prvt[\![x + y = z]\!])^{x}_{Sx}$	26 abv
28.	$\forall y \forall z(x + y = z \to Prvt[\![x + y = z]\!])^{x}_{Sx}$	27 ∀I
29.	$\forall y \forall z(x + y = z \to Prvt[\![x + y = z]\!]) \to \forall y \forall z(x + y = z \to Prvt[\![x + y = z]\!])^{x}_{Sx}$	16-28 →I
30.	$\forall y \forall z(x + y = z \to Prvt[\![x + y = z]\!])$	13,29 IN

We are able to apply the assumption to get $Prvt[\![x + y = z]\!]^{y}_{Sy}$ and convert this into the desired result. So PA $\vdash x + y = z \to Prvt[\![x + y = z]\!]$.

($\times$) Suppose $\mathscr{P}$ is $x \times y = z$. The proof in PA requires appeal to IN, on the value of x in $\forall y \forall z(x \times y = z \to Prvt[\![x \times y = z]\!])$. The zero case is straightforward. Then,

1.	$\forall y \forall z(x \times y = z \to Prvt[\![x \times y = z]\!])^{x}_{\emptyset}$	zero case
2.	$Sx \times y = z \leftrightarrow x \times y + y = z$	T6.58
3.	$x \times y = v \to (v + y = z \to x \times y + y = z)$	simple ND
4.	$Prvt[\![x \times y + y = z \to Sx \times y = z]\!]$	2 T13.52
5.	$Prvt[\![x \times y = v \to (v + y = z \to x \times y + y = z)]\!]$	3 T13.52
6.	$\forall y \forall z(x \times y = z \to Prvt[\![x \times y = z]\!])$	A (g →I)
7.	$(x \times y = z)^{x}_{Sx}$	A (g →I)
8.	$Sx \times y = z$	7 abv
9.	$x \times y + y = z$	2,8 ↔E
10.	$\exists v(x \times y = v)$	=I,∃I
11.	$x \times y = v$	A (g 10∃E)
12.	$v + y = z$	9,11 =E
13.	$Prvt[\![v + y = z]\!]$	12 (+) case
14.	$Prvt[\![x \times y = z]\!]^{z}_{v}$	6,11 ∀E,→E
15.	$Prvt[\![x \times y = v]\!]$	14 T13.54
16.	$Prvt[\![x \times y = v]\!] \to Prvt[\![v + y = z \to x \times y + y = v]\!]$	5 T13.53
17.	$Prvt[\![v + y = z \to x \times y + y = z]\!]$	15,16 →E
18.	$Prvt[\![v + y = z]\!] \to Prvt[\![x \times y + y = z]\!]$	17 T13.53
19.	$Prvt[\![x \times y + y = z]\!]$	18,13 →E
20.	$Prvt[\![x \times y + y = z]\!] \to Prvt[\![Sx \times y = z]\!]$	4 T13.53
21.	$Prvt[\![Sx \times y = z]\!]$	19,20 →E
22.	$Prvt[\![x \times y = z]\!]^{x}_{Sx}$	21 T13.54
23.	$Prvt[\![x \times y = z]\!]^{x}_{Sx}$	10,11-22 ∃E
24.	$(x \times y = z)^{x}_{Sx} \to Prvt[\![x \times y = z]\!]^{x}_{Sx}$	7-23 →I
25.	$(x \times y = z \to Prvt[\![x \times y = z]\!])^{x}_{Sx}$	24 abv
26.	$\forall y \forall z(x \times y = z \to Prvt[\![x \times y = z]\!])^{x}_{Sx}$	25 ∀I
27.	$\forall y \forall z(x \times y = z \to Prvt[\![x \times y = z]\!]) \to \forall y \forall z(x \times y = z \to Prvt[\![x \times y = z]\!])^{x}_{Sx}$	6-26 →I
28.	$\forall y \forall z(x \times y = z \to Prvt[\![x \times y = z]\!])$	1,27 IN

The previous result does not directly apply to $x \times y + y = z$. However, having identified $x \times y$ with variable v we get $Prvt[\![v + y = z]\!]$, and with the inductive assumption $Prvt[\![x \times y = v]\!]$. These then unpack into $Prvt[\![Sx \times y = z]\!]$. So PA $\vdash x \times y = z \to Prvt[\![x \times y = z]\!]$.

Assp: For any i, $0 \leq i < k$ if a $\Sigma^{\star}$ $\mathcal{P}$ has i operator symbols, then PA $\vdash \mathcal{P} \to Prvt[\![\mathcal{P}]\!]$.

Show: If a $\Sigma^{\star}$ $\mathcal{P}$ has k operator symbols, then PA $\vdash \mathcal{P} \to Prvt[\![\mathcal{P}]\!]$.

If $\Sigma^{\star}$ $\mathcal{P}$ has k operator symbols, then it is of the form, $\mathcal{A} \vee \mathcal{B}$, $\mathcal{A} \wedge \mathcal{B}$, $(\forall x \leq y)\mathcal{A}$ (y not in $\mathcal{A}$), or $\exists x\mathcal{A}$ for $\Sigma^{\star}$ $\mathcal{A}$ and $\mathcal{B}$ with $< k$ operator symbols.

($\wedge$) $\mathcal{P}$ is $\mathcal{A} \wedge \mathcal{B}$. Reason as follows.

1.	$\mathcal{A} \rightarrow Prvt[\![\mathcal{A}]\!]$	by assp
2.	$\mathcal{B} \rightarrow Prvt[\![\mathcal{B}]\!]$	by assp
3.	$\mathcal{A} \rightarrow (\mathcal{B} \rightarrow (\mathcal{A} \wedge \mathcal{B}))$	T9.4
4.	$Prvt[\![\mathcal{A} \rightarrow (\mathcal{B} \rightarrow (\mathcal{A} \wedge \mathcal{B}))]\!]$	3 T13.52
5.	$\quad \mathcal{A} \wedge \mathcal{B}$	A (g $\rightarrow$I)
6.	$\quad Prvt[\![\mathcal{A}]\!]$	1,5
7.	$\quad Prvt[\![\mathcal{B}]\!]$	2,5
8.	$\quad Prvt[\![\mathcal{A}]\!] \rightarrow Prvt[\![\mathcal{B} \rightarrow (\mathcal{A} \wedge \mathcal{B})]\!]$	4 T13.53
9.	$\quad Prvt[\![\mathcal{B} \rightarrow (\mathcal{A} \wedge \mathcal{B})]\!]$	6,8 $\rightarrow$E
10.	$\quad Prvt[\![\mathcal{B}]\!] \rightarrow Prvt[\![\mathcal{A} \wedge \mathcal{B}]\!]$	9 T13.53
11.	$\quad Prvt[\![\mathcal{A} \wedge \mathcal{B}]\!]$	7,10 $\rightarrow$E
12.	$(\mathcal{A} \wedge \mathcal{B}) \rightarrow Prvt[\![\mathcal{A} \wedge \mathcal{B}]\!]$	5-11 $\rightarrow$I

And similarly for $\vee$.

($\exists$) $\mathcal{P}$ is $\exists x \mathcal{A}$. Reason as follows.

1.	$\mathcal{A} \rightarrow Prvt[\![\mathcal{A}]\!]$	by assp
2.	$\mathcal{A} \rightarrow \exists x \mathcal{A}$	T3.29
3.	$Prvt[\![\mathcal{A} \rightarrow \exists x \mathcal{A}]\!]$	2 T13.52
4.	$\quad \exists x \mathcal{A}$	A (g $\rightarrow$I)
5.	$\quad\quad \mathcal{A}$	A (g 4$\exists$E)
6.	$\quad\quad Prvt[\![\mathcal{A}]\!]$	1,5 $\rightarrow$E
7.	$\quad\quad Prvt[\![\mathcal{A}]\!] \rightarrow Prvt[\![\exists x \mathcal{A}]\!]$	3 T13.53
8.	$\quad\quad Prvt[\![\exists x \mathcal{A}]\!]$	7,6 $\rightarrow$E
9.	$\quad Prvt[\![\exists x \mathcal{A}]\!]$	4,5-8 $\exists$E
10.	$\exists x \mathcal{A} \rightarrow Prvt[\![\exists x \mathcal{A}]\!]$	5-9 $\rightarrow$I

($\forall$) $\mathcal{P}$ is $(\forall x \leq y)\mathcal{A}$. The argument in PA requires appeal to IN, for induction on the value of y. For the zero case,

1.	$\mathcal{A}^x_\emptyset \to Prvt[\![\mathcal{A}^x_\emptyset]\!]$	by assp
2.	$(\forall x \leq \emptyset)\mathcal{A} \leftrightarrow \mathcal{A}^x_\emptyset$	thrm (with T8.21)
3.	$Prvt[\![\mathcal{A}^x_\emptyset \to (\forall x \leq \emptyset)\mathcal{A}]\!]$	2 T13.52
4.	$(\forall x \leq y)\mathcal{A}^y_\emptyset$	A ($g \to$I)
5.	$(\forall x \leq \emptyset)\mathcal{A}$	4 abv
6.	$\mathcal{A}^x_\emptyset$	2,5 $\leftrightarrow$E
7.	$Prvt[\![\mathcal{A}^x_\emptyset]\!]$	1,6 $\to$E
8.	$Prvt[\![\mathcal{A}^x_\emptyset]\!] \to Prvt[\![(\forall x \leq \emptyset)\mathcal{A}]\!]$	3 T13.53
9.	$Prvt[\![(\forall x \leq \emptyset)\mathcal{A}]\!]$	8,7 $\to$E
10.	$Prvt[\![(\forall x \leq y)\mathcal{A}^y_\emptyset]\!]$	9 abv
11.	$Prvt[\![(\forall x \leq y)\mathcal{A}]\!]^y_\emptyset$	10 T13.54
12.	$(\forall x \leq y)\mathcal{A}^y_\emptyset \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!]^y_\emptyset$	5-11 $\to$I
13.	$((\forall x \leq y)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!])^y_\emptyset$	12 abv

For (5) and (10) it is important that y in a bound quantifier of the $\Sigma^\star$ formula does not appear in $\mathcal{A}$. Now the inductive stage.

14.	$\mathcal{A}^x_{Sy} \to Prvt[\![\mathcal{A}^x_{Sy}]\!]$	by assp
15.	$(\forall x \leq Sy)\mathcal{A} \leftrightarrow (\forall x \leq y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}$	with T13.13n
16.	$Prvt[\![((\forall x \leq y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}) \to (\forall x \leq Sy)\mathcal{A}]\!]$	15 T13.52
17.	$(\forall x \leq y)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!]$	A ($g \to$I)
18.	$((\forall x \leq y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}) \to Prvt[\![(\forall x \leq y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}]\!]$	14,17 as for $\wedge$
19.	$(\forall x \leq Sy)\mathcal{A}$	A ($g \to$I)
20.	$(\forall x \leq y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}$	15,19 $\leftrightarrow$E
21.	$Prvt[\![(\forall x \leq y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}]\!]$	18,20 $\to$E
22.	$Prvt[\![(\forall x \leq y)\mathcal{A} \wedge \mathcal{A}^x_{Sy}]\!] \to Prvt[\![(\forall x \leq Sy)\mathcal{A}]\!]$	16 T13.53
23.	$Prvt[\![(\forall x \leq Sy)\mathcal{A}]\!]$	22,21 $\to$E
24.	$Prvt[\![(\forall x \leq y)\mathcal{A}]\!]^y_{Sy}$	23, T13.54
25.	$(\forall x \leq Sy)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!]^y_{Sy}$	19-24 $\to$I
26.	$((\forall x \leq y)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!])^y_{Sy}$	25 abv
27.	$((\forall x \leq y)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!]) \to ((\forall x \leq y)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!])^y_{Sy}$	17-26 $\to$I
28.	$(\forall x \leq y)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!]$	13,27 IN

So PA $\vdash (\forall x \leq y)\mathcal{A} \to Prvt[\![(\forall x \leq y)\mathcal{A}]\!]$.

Indct: For any $\Sigma^\star$ formula $\mathcal{P}$, PA $\vdash \mathcal{P} \to Prvt[\![\mathcal{P}]\!]$.

Now it is a simple matter to pull together our results into the third derivability condition.

T13.60. For any formula $\mathcal{P}$, PA $\vdash \Box\mathcal{P} \to \Box\Box\mathcal{P}$

Consider any formula $\mathcal{P}$ and the Σ_1 sentence $\Box\mathcal{P}$. By T13.58, there is a $(\Box\mathcal{P})^\star$ such that PA $\vdash \Box\mathcal{P} \leftrightarrow (\Box\mathcal{P})^\star$. By T13.59, PA $\vdash (\Box\mathcal{P})^\star \rightarrow \mathit{Prvt}[\![(\Box\mathcal{P})^\star]\!]$. Reason as follows.

1.	$(\Box\mathcal{P})^\star \rightarrow \mathit{Prvt}[\![(\Box\mathcal{P})^\star]\!]$	T13.59
2.	$\Box\mathcal{P} \leftrightarrow (\Box\mathcal{P})^\star$	T13.58
3.	$\mathit{Prvt}[\![(\Box\mathcal{P})^\star \rightarrow \Box\mathcal{P}]\!]$	2 T13.52
4.	$\mathit{Prvt}[\![(\Box\mathcal{P})^\star]\!] \rightarrow \mathit{Prvt}[\![\Box\mathcal{P}]\!]$	3 T13.53
5.	$\Box\mathcal{P} \rightarrow \mathit{Prvt}[\![\Box\mathcal{P}]\!]$	2,1,4 HS

So PA $\vdash \Box\mathcal{P} \rightarrow \mathit{Prvt}[\![\Box\mathcal{P}]\!]$; and since $\Box\mathcal{P}$ is a sentence, this is to say, PA $\vdash \Box\mathcal{P} \rightarrow \mathit{Prvt}(\overline{\ulcorner\Box\mathcal{P}\urcorner})$; which is to say, PA $\vdash \Box\mathcal{P} \rightarrow \Box\Box\mathcal{P}$.

So, at long last, we have a demonstration of D3 and so, given deomonstration of the other conditions, of Gödel's second incompleteness theorem.

E13.39. Complete the demonstration of T13.59 by completing the remaining cases.

13.6 Reflections on the theorem

We conclude this chapter with a couple final reflections and consequences on our results.

13.6.1 Consistency sentences

As is typically done, we have let *Cont* be $\sim\mathit{Prvt}(\overline{\ulcorner\emptyset = S\emptyset\urcorner})$. If T is inconsistent, then T proves anything, so $T \vdash \overline{0} = \overline{1}$. And, supposing T extends Q, $T \vdash \overline{0} \neq \overline{1}$; so if $T \vdash \overline{0} = \overline{1}$, then T is inconsistent. But other sentences would do as well. So, where $\mathcal{T}$ is any theorem of T, we might let Cont' be $\sim\mathit{Prvt}(\overline{\ulcorner\sim\mathcal{T}\urcorner})$. In particular, we might simply consider the case where $\sim\mathcal{T}$ is (equivalent to) $\perp$ and set $\mathit{Cont}' = \sim\mathit{Prvt}(\overline{\ulcorner\perp\urcorner})$. (Where $\perp$ is $Z \wedge \sim Z$, it is equivalent to the negation of the theorem, $\sim(Z \wedge \sim Z)$). Then it is easy to see that PA $\vdash \mathit{Cont} \leftrightarrow \mathit{Cont}'$.

> PA $\vdash \emptyset = S\emptyset \leftrightarrow \perp$; so with D1, PA $\vdash \mathit{Prvt}(\overline{\ulcorner\emptyset = S\emptyset \leftrightarrow \perp\urcorner})$; so with D2, PA $\vdash \mathit{Prvt}(\overline{\ulcorner\emptyset = S\emptyset\urcorner}) \leftrightarrow \mathit{Prvt}(\overline{\ulcorner\perp\urcorner})$; and contraposing, PA $\vdash \mathit{Cont} \leftrightarrow \mathit{Cont}'$.

Again, one might let $\mathit{Cont}'' = \sim\exists x(\mathit{Prvt}(x) \wedge \overline{\mathit{Prvt}}(x))$, where $\overline{\mathit{Prvt}}(x))$ just in case there is a proof of the negation of the formula with Gödel number x. Then T is consistent just in case there is no proof of a formula and its negation. Again, PA $\vdash \mathit{Cont} \leftrightarrow \mathit{Cont}''$. This time the result requires a bit more work.

First, since a contradiction implies anything, PA $\vdash \emptyset = S\emptyset \to A$ and PA $\vdash \emptyset = S\emptyset \to \sim A$; reason as follows.

1.	$\emptyset = S\emptyset \to A$	thrm
2.	$\emptyset = S\emptyset \to \sim A$	thrm
3.	$Prvt(\overline{\ulcorner \emptyset = S\emptyset \to A \urcorner})$	1 D1
4.	$Prvt(\overline{\ulcorner \emptyset = S\emptyset \to \sim A \urcorner})$	2 D1
5.	$\quad Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	A (g →I)
6.	$\quad Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \to Prvt(\overline{\ulcorner A \urcorner})$	3 D2
7.	$\quad Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \to Prvt(\overline{\ulcorner \sim A \urcorner})$	4 D2
8.	$\quad Prvt(\overline{\ulcorner A \urcorner}) \wedge Prvt(\overline{\ulcorner \sim A \urcorner})$	5,6,7
9.	$\quad \exists x(Prvt(x) \wedge \overline{Prvt}(x))$	8 ∃I
10.	$Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \to \exists x(Prvt(x) \wedge \overline{Prvt}(x))$	7-9 →I

So PA $\vdash Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \to \exists x(Prvt(x) \wedge \overline{Prvt}(x))$.

The other direction is not much more difficult. Insofar as the antecedent is existentially quantified we shall not be able to depend on capture for any particular sentence. However, reasoning very much as for T13.47, it is not hard to show,

T13.61. PA $\vdash \mathbb{A}xiom(p) \to \mathbb{P}rvt(p)$

Where $\mathcal{A}_1 \ldots \mathcal{A}_n$ are the elements of some sentential form $\mathcal{P}$, let $\mathcal{A}^*$ be a for variable a; $\sim\mathcal{P}^*$ be $\mathbb{n}eg(p)$; and $(\mathcal{P} \to \mathcal{Q})^*$, be $\mathbb{c}nd(p, q)$. Then where PA $\vdash \mathcal{P}$, we shall be able to show PA $\vdash \mathbb{W}ff(a) \wedge \ldots \wedge \mathbb{W}ff(b) \to \mathbb{P}rvt(\mathcal{P}^*)$. The reasoning is by a simple induction (of a sort we have seen before): Given an AD derivation of $\mathcal{P}$, under the assumption $\mathbb{W}ff(a) \wedge \ldots \wedge \mathbb{W}ff(b)$, corresponding to any axiom $\mathcal{A}$, we may use the definition to get $\mathbb{A}xiom(\mathcal{A}^*)$ and then T13.61 for $\mathbb{P}rvt(\mathcal{A}^*)$. Corresponding to an application of MP to some $\mathcal{P}$ and $\mathcal{P} \to \mathcal{Q}$, use T13.47 to convert $\mathbb{P}rvt(\mathbb{c}nd(\mathcal{P}^*, \mathcal{Q}^*))$ to $\mathbb{P}rvt(\mathcal{P}^*) \to \mathbb{P}rvt(\mathcal{Q}^*)$ and apply MP. As an example, compare the following lines of the sort we might have obtained in chapter 3,

1.	$A \to (B \to A)$	A1
2.	$[A \to (B \to A)] \to [(A \to B) \to (A \to A)]$	A2
3.	$(A \to B) \to (A \to A)$	1,2 MP

and the derived version,

0.	$\mathbb{W}ff(a) \wedge \mathbb{W}ff(b)$	A
1.1.	$\mathbb{A}xiom(\mathbb{cnd}(a, \mathbb{cnd}(b, a)))$	1 def
1.	$\mathbb{P}rvt(\mathbb{cnd}(a, \mathbb{cnd}(b, a)))$	1.1 T13.61
2.1.	$\mathbb{A}xiom(\mathbb{cnd}(\mathbb{cnd}[a, \mathbb{cnd}(b, a)], \mathbb{cnd}[\mathbb{cnd}(a, b), \mathbb{cnd}(a, a)]))$	1 def
2.	$\mathbb{P}rvt(\mathbb{cnd}(\mathbb{cnd}[a, \mathbb{cnd}(b, a)], \mathbb{cnd}[\mathbb{cnd}(a, b), \mathbb{cnd}(a, a)]))$	2.1 T13.61
3.1.	$\mathbb{P}rvt(\mathbb{cnd}[a, \mathbb{cnd}(b, a)]) \to \mathbb{P}rvt(\mathbb{cnd}[\mathbb{cnd}(a, b), \mathbb{cnd}(a, a)])$	2 T13.47
3.	$\mathbb{P}rvt(\mathbb{cnd}[\mathbb{cnd}(a, b), \mathbb{cnd}(a, a)])$	1,3.1 MP
4.	$\mathbb{W}ff(a) \wedge \mathbb{W}ff(b) \to \mathbb{P}rvt(\mathbb{cnd}[\mathbb{cnd}(a, b), \mathbb{cnd}(a, a)])$	0 - 3 DT

And similarly we might show the correlate to T3.9, $\vdash \sim\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$, which we record as a theorem.

T13.62. PA $\vdash \mathbb{W}ff(a) \wedge \mathbb{W}ff(b) \to \mathbb{P}rvt(\mathbb{cnd}[\mathbb{neg}(a), \mathbb{cnd}(a, b)])$.

But then we may reason as follows.

1.	$\mathbb{W}ff(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	cap
2.	$\exists x[Prvt(x) \wedge \overline{Prvt}(x)]$	A (g $\to$I)
3.	$Prvt(j) \wedge \overline{Prvt}(j)$	A (g 2$\exists$E)
4.	$\mathbb{W}ff(j)$	3 T13.46k
5.	$\mathbb{P}rvt(\mathbb{cnd}[\mathbb{neg}(j), \mathbb{cnd}(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner})])$	1,4 T13.62
6.	$\mathbb{P}rvt(\mathbb{neg}(j)) \to \mathbb{P}rvt(\mathbb{cnd}(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner}))$	5 T13.47
7.	$\mathbb{P}rvt(\mathbb{cnd}(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner}))$	3,6 $\wedge$E,$\to$E
8.	$\mathbb{P}rvt(j) \to \mathbb{P}rvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	7 T13.47
9.	$Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	3,8 $\wedge$E,$\to$E
10.	$Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	2,3-9 $\exists$E
11.	$\exists x[Prvt(x) \wedge \overline{Prvt}(x)] \to Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$	2-10 $\to$I

Again note the requirement that we reason with free variables under the assumption for $\exists$E.

Putting the parts together, PA $\vdash Prvt(\overline{\ulcorner \emptyset = S\emptyset \urcorner}) \leftrightarrow \exists x(Prvt(x) \wedge \overline{Prvt}(x))$; and contraposing, PA $\vdash Cont \leftrightarrow Cont''$. So, to this extent, it does not matter which version of the consistency statement we select. Underlying the point that these different statements are equivalent is that anything follows from a contradiction — so that the one follows from the others.[11]

Having proved PA $\nvdash$ *Cont*, we have PA $\nvdash Cont'$ and PA $\nvdash Cont''$. These are particular sentences which, like $\mathscr{G}$, are unprovable. And, now that we have the

[11]This equivalence breaks down in a non-classical logic which blocks *ex falso quodlibet*, the principle that from a contradiction anything follows. So, for example, in relevant logic, it might be that there is some A such that $T \vdash A \wedge \sim A$ but $T \nvdash \emptyset = S\emptyset$. See Priest, *Non-Classical Logics* for an introduction to these matters.

derivability conditions, with T13.11, neither are their negations provable. They have special interest because because each "says" that PA is consistent. Still, it is worth asking whether there is some different sentence to express the consistency of PA such that *it* would be provable. Consider, for example a trick related to the Rosser sentence,

$$Prft^c(x,y) =_{\text{def}} Prft(x,y) \wedge (\forall v \leq x){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$$

So $Prft^c(x,y)$ requires a measure of consistency: it says x numbers a proof of the formula numbered y and no proof numbered less than x demonstrates inconsistency ($\emptyset = \overline{1}$). Then so long as PA *is* consistent $Prft^c(x,y)$ continues to capture $\textsc{prft}(\mathsf{x},\mathsf{y})$.

(i) Suppose $\langle \mathsf{m}, \mathsf{n} \rangle \in \textsc{prft}$. (a) By capture, PA $\vdash Prft(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. And (b), since PA is consistent, there is no proof of a contradiction in PA and again by capture, PA $\vdash {\sim}Prft(\overline{0}, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$; PA $\vdash {\sim}Prft(\overline{1}, \overline{\ulcorner \emptyset = S\emptyset \urcorner})\ldots$ and PA $\vdash {\sim}Prft(\overline{\mathsf{m}}, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$; so with T8.21, PA $\vdash (\forall v \leq \overline{\mathsf{m}}){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$; so PA $\vdash Prft^c(\overline{\mathsf{m}}, \overline{\mathsf{n}})$.

(ii) Suppose $\langle \mathsf{m}, \mathsf{n} \rangle \notin \textsc{prft}$; then by capture, PA $\vdash {\sim}Prft(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. So PA $\vdash {\sim}[Prft(\overline{\mathsf{m}}, \overline{\mathsf{n}}) \wedge (\forall v \leq \overline{\mathsf{m}}){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})]$, which is to say PA $\vdash {\sim}Prft^c(\overline{\mathsf{m}}, \overline{\mathsf{n}})$.

And, with T12.6, $Prft^c(x,y)$ expresses $\textsc{prft}(\mathsf{x},\mathsf{y})$ as well. Given this, set $Prvt^c(y) =_{\text{def}} \exists x Prft^c(x,y)$, and $Cont^c =_{\text{def}} {\sim}Prvt^c(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$. The idea, then is that $Cont^c$ just in case PA is consistent.

But $Prvt^c$ is designed so that $Prvt^c(\overline{\ulcorner \emptyset = S\emptyset \urcorner})$ is impossible — and it is easy to see that $Cont^c$ is therefore provable.

1.	$\exists x[Prft(x, \overline{\ulcorner \emptyset = S\emptyset \urcorner}) \wedge (\forall v \leq x){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})]$	A (c, $\sim$I)
2.	$Prft(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner}) \wedge (\forall v \leq j){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$	A (c 1$\exists$E)
3.	$Prft(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$	2 $\wedge$E
4.	$(\forall v \leq j){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$	2 $\wedge$E
5.	$j \leq j$	with T13.13l
6.	${\sim}Prft(j, \overline{\ulcorner \emptyset = S\emptyset \urcorner})$	4,5 ($\forall$E)
7.	$\perp$	3,6 $\perp$I
8.	$\perp$	1,2-7 $\exists$E
9.	${\sim}\exists x[Prft(x, \overline{\ulcorner \emptyset = S\emptyset \urcorner}) \wedge (\forall v \leq x){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})]$	1-8 $\sim$I

So PA $\vdash {\sim}\exists x[Prft(x, \overline{\ulcorner \emptyset = S\emptyset \urcorner}) \wedge (\forall v \leq x){\sim}Prft(v, \overline{\ulcorner \emptyset = S\emptyset \urcorner})]$ which is to say PA $\vdash Cont^c$. This is because $Prft^c$ builds in from the start that nothing numbers a proof of $\emptyset = S\emptyset$.

Intuitively, so long as PA is consistent, $Prft^c$ works just fine. But if PA is not consistent, then it no longer tracks with proof. Similarly, if PA is consistent, $Cont^c$ plausibly "says" PA is consistent. But if PA is inconsistent then it no longer tracks with consistency. So its provability is, in this sense, uninteresting.

Insofar as $Cont^c$ is provable it must be that $Prvt^c$ fails one or more of the derivability conditions. To see how this might be, consider D2 and suppose PA is inconsistent and proofs are ordered according to their Gödel numbers as follows,

$$\mathcal{A} \to \mathcal{B} \qquad\qquad \mathcal{A} \qquad\qquad \emptyset = S\emptyset \qquad\qquad \mathcal{B}$$

Then PA $\vdash Prvt(\overline{\ulcorner \mathcal{B} \urcorner})$ but, insofar as the proof of $\mathcal{B}$ is numbered greater than the proof of $\emptyset = S\emptyset$, PA $\vdash \sim Prvt^c(\overline{\ulcorner \mathcal{B} \urcorner})$. In this case, D2 fails, so that our main argument to show PA $\nvdash$ *Cont* does not apply to $Cont^c$.

13.6.2 Löb's Theorem

If T is a recursively axiomatized theory extending Q, by the diagonal lemma there is a sentence $\mathcal{H}$, of which $\mathcal{G}$ is a sample, such that $T \vdash \mathcal{H} \leftrightarrow \sim Prvt(\overline{\ulcorner \mathcal{H} \urcorner})$ — that is, $T \vdash \mathcal{H} \leftrightarrow \sim\Box\mathcal{H}$. We have seen that such a formula $\mathcal{H}$ is not provable. But, of course, by the diagonal lemma, there is another sentence $\mathcal{H}$ such that $T \vdash \mathcal{H} \leftrightarrow \Box\mathcal{H}$. In a brief note, "A Problem Concerning Provability" L. Henkin asks whether this $\mathcal{H}$ is provable. Supposing the first is analogous to the liar, 'this sentence is not true', the latter is like the truth-teller, 'this sentence is true'. An answer to Henkin's question follows immediately from Löb's theorem.

T13.63. Suppose T is a recursively axiomatized theory for which the derivability conditions D1 - D3 hold and $T \vdash \Box\mathcal{P} \to \mathcal{P}$, then $T \vdash \mathcal{P}$. *Löb's Theorem.*

Suppose T is a recursively axiomatized theory for which the derivability conditions hold and $T \vdash \Box\mathcal{P} \to \mathcal{P}$. Then the diagonal lemma obtains as well. Consider $Prvt(y) \to \mathcal{P}$; this is an expression of the sort $\mathcal{F}(y)$ to which the diagonal lemma applies; so by the diagonal lemma there is some $\mathcal{H}$ such that, $T \vdash \mathcal{H} \leftrightarrow (Prvt(\overline{\ulcorner \mathcal{H} \urcorner}) \to \mathcal{P})$ — that is, $T \vdash \mathcal{H} \leftrightarrow (\Box\mathcal{H} \to \mathcal{P})$. Now reason as follows.

1.	$\Box\mathcal{P} \rightarrow \mathcal{P}$	P
2.	$\mathcal{H} \leftrightarrow (\Box\mathcal{H} \rightarrow \mathcal{P})$	diag lemma
3.	$[\mathcal{H} \rightarrow (\Box\mathcal{H} \rightarrow \mathcal{P})] \wedge [(\Box\mathcal{H} \rightarrow \mathcal{P}) \rightarrow \mathcal{H}]$	2 abv
4.	$\mathcal{H} \rightarrow (\Box\mathcal{H} \rightarrow \mathcal{P})$	3 with T3.20
5.	$\Box[\mathcal{H} \rightarrow (\Box\mathcal{H} \rightarrow \mathcal{P})]$	4 D1
6.	$\Box[\mathcal{H} \rightarrow (\Box\mathcal{H} \rightarrow \mathcal{P})] \rightarrow [\Box\mathcal{H} \rightarrow \Box(\Box\mathcal{H} \rightarrow \mathcal{P})]$	D2
7.	$\Box\mathcal{H} \rightarrow \Box(\Box\mathcal{H} \rightarrow \mathcal{P})$	6,5 MP
8.	$\Box(\Box\mathcal{H} \rightarrow \mathcal{P}) \rightarrow (\Box\Box\mathcal{H} \rightarrow \Box\mathcal{P})$	D2
9.	$\Box\mathcal{H} \rightarrow (\Box\Box\mathcal{H} \rightarrow \Box\mathcal{P})$	7,8 T3.2
10.	$[\Box\mathcal{H} \rightarrow (\Box\Box\mathcal{H} \rightarrow \Box\mathcal{P})] \rightarrow [(\Box\mathcal{H} \rightarrow \Box\Box\mathcal{H}) \rightarrow (\Box\mathcal{H} \rightarrow \Box\mathcal{P})]$	A2
11.	$(\Box\mathcal{H} \rightarrow \Box\Box\mathcal{H}) \rightarrow (\Box\mathcal{H} \rightarrow \Box\mathcal{P})$	10,9 MP
12.	$\Box\mathcal{H} \rightarrow \Box\Box\mathcal{H}$	D3
13.	$\Box\mathcal{H} \rightarrow \Box\mathcal{P}$	11,12 MP
14.	$\Box\mathcal{H} \rightarrow \mathcal{P}$	13,1 T3.2
15.	$(\Box\mathcal{H} \rightarrow \mathcal{P}) \rightarrow \mathcal{H}$	3 with T3.19
16.	$\mathcal{H}$	15,14 MP
17.	$\Box\mathcal{H}$	16 D1
18.	$\mathcal{P}$	14,17 MP

So $T \vdash \mathcal{P}$. Now return to our original question. Suppose $T \vdash \mathcal{H} \leftrightarrow \Box\mathcal{H}$; then $T \vdash \Box\mathcal{H} \rightarrow \mathcal{H}$; so by Löb's theorem, $T \vdash \mathcal{H}$. So if T proves $\mathcal{H} \leftrightarrow \Box\mathcal{H}$, then T proves $\mathcal{H}$. Observe that in the presence of incompleteness, it must be the case that there is some sentence $\mathcal{L}$ such that $T \nvdash \mathcal{L}$, so that $T \nvdash \Box\mathcal{L} \rightarrow \mathcal{L}$. But for a sound theory, any sentence $\Box\mathcal{L} \rightarrow \mathcal{L}$ must be true; so here is another sentence true, but not provable.

Final theorems of chapter 13

T13.22 Where $\mathcal{F}(\vec{x}, y, v)$ is the formula for recursion, PA $\vdash \forall m \forall n[(\mathcal{F}(\vec{x}, y, m) \land \mathcal{F}(\vec{x}, y, n)) \rightarrow m = n]$.

T13.23 Results for $a \dot{-} b$. T13.24 results for $a|b$. T13.25 results for $Pr(a)$ and $Rp(a)$. T13.26 results for $lcm(a)$.

T13.27 PA $\vdash [(\forall i < k)(m(i) > \emptyset \land m(i) > h(i)) \land \forall i \forall j(i < j \land j < k \rightarrow Rp(Sm(i), Sm(j)))] \rightarrow \exists p(\forall i < k)rm(p, m(i)) = h(i)$ (*CRT*).

T13.28 Results for *maxp* and *maxs*.

T13.29 PA $\vdash \exists p \exists q(\forall i < k)\beta(p, q, i) = h(i)$.

T13.30 PA $\vdash \exists p \exists q[(\forall i < k)\beta(p, q, i) = \beta(r, s, i) \land \beta(p, q, k) = n]$.

T13.31 PA $\vdash \exists v \exists p \exists q[\beta(p, q, \emptyset) = g(\vec{x}) \land (\forall i < y)h(\vec{x}, i, \beta(p, q, i)) = \beta(p, q, Si) \land \beta(p, q, y) = v]$.

T13.32 For any friendly recursive relation $\mathsf{R}(\vec{\mathsf{x}})$ with characteristic function $\mathsf{ch_R}(\vec{\mathsf{x}})$, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathbb{ch}_{\mathsf{R}}(\vec{x}) = \emptyset$. And for a recursive operator $\mathsf{OP}(\mathsf{P_1}(\vec{\mathsf{x}}) \ldots \mathsf{P_n}(\vec{\mathsf{x}}))$ with characteristic function $\mathsf{f}(\mathsf{ch_{P_1}}(\vec{\mathsf{x}}) \ldots \mathsf{ch_{P_n}}(\vec{\mathsf{x}}))$, PA $\vdash \mathbb{O}p(\mathbb{P}_1(\vec{x}) \ldots \mathbb{P}_n(\vec{x})) \leftrightarrow \mathbb{f}(\mathbb{ch}_{\mathsf{P_1}}(\vec{x}) \ldots \mathbb{ch}_{\mathsf{P}_n}(\vec{x})) = \emptyset$. Corollary: where $\mathsf{R}(\vec{\mathsf{x}})$ is originally captured by $\mathcal{R}(\vec{x}, \emptyset)$, PA $\vdash \mathbb{R}(\vec{x}) \leftrightarrow \mathcal{R}(\vec{x}, \emptyset)$.

T13.33 Suppose $\mathbb{f}(\vec{x}, y)$ is defined by $\mathbb{g}(\vec{x})$ and $\mathbb{h}(\vec{x}, y, u)$ so that PA $\vdash v = \mathbb{f}(\vec{x}, y) \leftrightarrow \mathcal{F}(\vec{x}, y, v)$; then, (i) $\mathbb{f}(\vec{x}, \emptyset) = \mathbb{g}(\vec{x})$ and (ii) $\mathbb{f}(\vec{x}, S(y)) = \mathbb{h}(\vec{x}, y, \mathbb{f}(\vec{x}, y))$.

T13.34 Equivalences for $\mathbb{suc}$, $\mathbb{zero}$, $\mathbb{idnt}^j_k$, $\mathbb{plus}$ and $\mathbb{times}$. T13.35 results for *pred*, *sg* and *csg*. T13.36 Equivalences for $\mathbb{pred}$, $\mathbb{subc}$, $\mathbb{absval}$, $\mathbb{sg}$, $\mathbb{csg}$, $\mathbb{Eq}$, $\mathbb{Leq}$, $\mathbb{Less}$, $\mathbb{Neg}$, and $\mathbb{Dsj}$. T13.37 PA proves a characteristic function takes the value $\emptyset$ or $\overline{1}$. T13.38 Equivalences for $(\exists y \leq z)$, $(\exists y < z)$, $(\forall y \leq z)$, $(\forall y < z)$, $(\mu y \leq z)$, $\mathbb{Fctr}$, and $\mathbb{Prime}$.

T13.39 First applications to recursive functions.

T13.40 Results for m^a. T13.41 results for $\mathbb{fact}$. T13.42 results for $\mathbb{pi}$. T13.43 results for $\mathbb{exp}$. T13.44 results for $\mathbb{len}$. T13.45 results for $m * n$.

T13.47 PA $\vdash \Box(\mathcal{P} \rightarrow \mathcal{Q}) \rightarrow (\Box\mathcal{P} \rightarrow \Box\mathcal{Q})$.

T13.52 If PA $\vdash \mathcal{P}$, then PA $\vdash Prvt[\![\mathcal{P}]\!]$ — analog to D1

T13.53 PA $\vdash Prvt[\![\mathcal{P} \rightarrow \mathcal{Q}]\!] \rightarrow (Prvt[\![\mathcal{P}]\!] \rightarrow Prvt[\![\mathcal{Q}]\!])$ — analog to D2

T13.54 If t is one of $\emptyset$, y or Sy, then PA $\vdash Prvt[\![\mathcal{P}^x_t]\!] \leftrightarrow Prvt[\![\mathcal{P}]\!]^x_t$.

T13.55 For any $\mathcal{P}$ of the form $t = x$, there is a $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

T13.56 For any Δ_0 formula $\mathcal{P}$, there is a normal formula $\mathcal{P}^*$ such that $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^*$.

T13.57 For any Δ_0 formula $\mathcal{P}$ there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

T13.58 For any Σ_1 formula $\mathcal{P}$ there is a $\Sigma^\star$ formula $\mathcal{P}^\star$ such that PA $\vdash \mathcal{P} \leftrightarrow \mathcal{P}^\star$.

T13.59 For any $\Sigma^\star$ formula $\mathcal{P}$, PA $\vdash \mathcal{P} \rightarrow Prvt[\![\mathcal{P}]\!]$.

T13.60 For any formula $\mathcal{P}$, PA $\vdash \Box\mathcal{P} \rightarrow \Box\Box\mathcal{P}$

T13.63 Suppose T is a recursively axiomatized theory for which the derivability conditions D1 - D3 hold and $T \vdash \Box\mathcal{P} \rightarrow \mathcal{P}$, then $T \vdash \mathcal{P}$. *Löb's Theorem.*

Chapter 14

Logic and Computability

In this chapter, we begin with the notion of a Turing machine, and a Turing computable function. It turns out that the Turing computable functions are the same as the recursive functions. Once we have seen this, it is a short step from a problem about computability — the *halting problem*, to another demonstration of essential results. Further, according to Church's thesis, the Turing computable functions, and so the recursive functions, are *all* the algorithmically computable functions. This converts results like T12.22 according to which no recursive function is true just of (numbers for) theorems of predicate logic, into ones according to which no algorithmically computable function is true just of theorems of predicate logic — where this result is much more than a curiosity about an obscure class of functions.

14.1 Turing Computable Functions

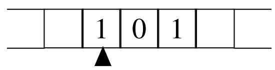
We begin saying what a Turing machine, and the Turing computable functions are. Then we turn to demonstrations that Turing computable functions are recursive, and recursive functions are Turing computable.

14.1.1 Turing Machines

A Turing machine is a simple device which, despite its simplicity, is capable of computing any recursive function — and capable of computing whatever is computable by the more sophisticated computers with which we are familiar.[1]

[1]So called after Alan Turing, who originally proposed them hypothetically, prior to the existence of modern computing devices, for purposes much like our own. Turing went on to develop electromechanical machines for code breaking during World War II, and was involved in development of early

We may think of a Turing machine as consisting of a *tape*, *machine head*, and a finite set of *instruction quadruples*.[2]

(A) 

The tape is a sequence of cells, infinite in two directions, where the cells may be empty or filled with $\mathsf{0}$ or $\mathsf{1}$. The machine head, indicated by arrow, reads or writes the contents of a given cell, and moves left or right, one cell at a time. The head is capable of five actions: (L) move left one cell; (R) move right one cell; (B) write a blank; ($\mathsf{0}$) write a zero; ($\mathsf{1}$) write a one. When the head is over a cell it is capable of reading or writing the contents of that cell.

Instruction quadruples are of the sort, $\langle \mathsf{q}_1, \mathsf{C}, \mathsf{A}, \mathsf{q}_2 \rangle$ and constitute a function in the sense that no two quadruples have $\langle \mathsf{q}_1, \mathsf{C} \rangle$ the same but $\langle \mathsf{A}, \mathsf{q}_2 \rangle$ different. For an instruction quadruple: (q_1) labels the quadruple; (C) is a possible state or content of the scanned cell; (A) is one of the five actions; (q_2) is a label for some (other) quadruples. In effect, an instruction quadruple q_1 says, "if the current cell has content C, perform action A and go to instruction q_2." The machine begins at an instruction with label $\mathsf{q}_1 = \mathsf{1}$, and stops when $\mathsf{q}_2 = \mathsf{0}$.

For a simple example, consider the following quadruples, along with the tape (A) from above.

	$\langle \mathsf{1}, \mathsf{0}, \mathsf{R}, \mathsf{1} \rangle$	if $\mathsf{0}$ move right
(B)	$\langle \mathsf{1}, \mathsf{1}, \mathsf{0}, \mathsf{1} \rangle$	if $\mathsf{1}$ write $\mathsf{0}$
	$\langle \mathsf{1}, \mathsf{B}, \mathsf{L}, \mathsf{2} \rangle$	end of word, back up and go to instruction $\mathsf{2}$
	$\langle \mathsf{2}, \mathsf{0}, \mathsf{L}, \mathsf{2} \rangle$	while value is $\mathsf{0}$, move left
	$\langle \mathsf{2}, \mathsf{B}, \mathsf{R}, \mathsf{0} \rangle$	end of word, return right and stop

The machine begins at label $\mathsf{1}$. In this case, the head is over a cell with content $\mathsf{1}$; so from the second instruction the machine writes $\mathsf{0}$ in that cell and returns to instruction label $\mathsf{1}$. Because the cell now contains $\mathsf{0}$, the machine reads $\mathsf{0}$; so, from instruction $\mathsf{1}$, the head moves right one space and returns to instruction $\mathsf{1}$ again. Now the machine reads $\mathsf{0}$; so it moves right again and goes returns to instruction $\mathsf{1}$. Because it reads $\mathsf{1}$, again the machine writes $\mathsf{0}$ and goes to instruction $\mathsf{1}$ where it moves right and goes to $\mathsf{1}$. Now the head is over a blank; so it moves left one cell, and goes to $\mathsf{2}$. At instruction $\mathsf{2}$, the head moves left so long as the tape reads $\mathsf{0}$. When the head reaches a blank, it moves right one space, back over the word, and stops. So the result is,

stored-program computers after the war.

[2]Specifications of Turing machines differ somewhat. So, for example, some versions allow instruction quintuples, and allow different symbols on the tape. Nothing about what is computable changes on the different accounts.

(C)

		0	0	0		
		▲				

In the standard case, we begin with a blank tape except for one or more binary "words" where the words are separated by single blank cells, and the machine head is over the left-most cell of the left-most block. The above example is a simple case of this sort, but also,

(D)

		1	0	1	1		1	1	0	1	0		1		
		▲													

And in the usual case the program halts with the head over the leftmost cell of a single word on the tape. A function $\mathsf{f}(\vec{\mathsf{x}})$ is *Turing computable* when, beginning with $\vec{\mathsf{x}}$ on the tape in binary digits, the result is $\mathsf{f}(\vec{\mathsf{x}})$. Thus our little program computes $\mathsf{zero}(\mathsf{x})$, beginning with any x, and returning the value $\mathsf{0}$.

It will be convenient to require that programs are *dextral* (right-handed), in the sense that (a) in executing a program we never write in a cell to the left of the initial cell, or scan a cell more than one to the left of the initial cell; and (b) when the program halts, the head is over the initial cell and the final result begins in the same cell as the initial scanned cell. This does not affect what can be computed, but aids in predicting results when Turing programs are combined. Our little program is dextral.

A program to compute $\mathsf{suc}(\mathsf{x})$ is not much more difficult. Let us begin by thinking about what we want the program to do. With a three-digit input word, the desired outputs are,

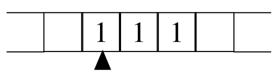
$$\begin{array}{lclclcl}
\mathsf{000} & \Longrightarrow & \mathsf{001} & \quad & \mathsf{100} & \Longrightarrow & \mathsf{101} \\
\mathsf{001} & \Longrightarrow & \mathsf{010} & & \mathsf{101} & \Longrightarrow & \mathsf{110} \\
\mathsf{010} & \Longrightarrow & \mathsf{011} & & \mathsf{110} & \Longrightarrow & \mathsf{111} \\
\mathsf{011} & \Longrightarrow & \mathsf{100} & & \mathsf{111} & \Longrightarrow & \mathsf{1000}
\end{array}$$

Moving from the right of the input word, we want to turn any one to a zero until we can turn a zero (or a blank) to a one. Here is a way to do that.

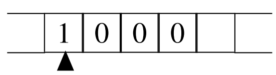
	$\langle 1, 0, R, 1\rangle$	move to end of word
(E)	$\langle 1, 1, R, 1\rangle$	
	$\langle 1, B, L, 5\rangle$	
	$\langle 5, 0, 1, 7\rangle$	flip 1 to 0 then 0 or blank to 1
	$\langle 5, 1, 0, 6\rangle$	
	$\langle 5, B, 1, 7\rangle$	
	$\langle 6, 0, L, 5\rangle$	
	$\langle 7, 0, L, 7\rangle$	return to start
	$\langle 7, 1, L, 7\rangle$	
	$\langle 7, B, R, 0\rangle$	

Do not worry about the gap in instruction labels. Nothing so-far requires instruction labels be sequential. This program moves the head to the right end of the word; from the right, flips one to zero until it finds a zero or blank; once it has acted on a zero or blank, it returns to the start.

So-far, so-good. But there is a problem with this program: In the case when the input is, say,

(F) 

the output is,



with the first symbol one to the left of the initial position. We turn the first blank to the left of the initial position to a one. So the program is not dextral. The problem is solved by "shifting" the word in the case when it is all ones.

	if solid ones shift right	flip 1 to 0 then 0 to 1
	$\langle 1, 0, R, 4\rangle$	$\langle 5, 0, 1, 7\rangle$
	$\langle 1, 1, R, 1\rangle$	$\langle 5, 1, 0, 6\rangle$
	$\langle 1, B, 1, 2\rangle$	$\langle 5, B, 1, 7\rangle$
	$\langle 2, 1, L, 2\rangle$	$\langle 6, 0, L, 5\rangle$
	$\langle 2, B, R, 3\rangle$	
(G)		return to start
	$\langle 3, 1, B, 3\rangle$	$\langle 7, 0, L, 7\rangle$
	$\langle 3, B, R, 4\rangle$	$\langle 7, 1, L, 7\rangle$
		$\langle 7, B, R, 0\rangle$
	$\langle 4, 0, R, 4\rangle$	
	$\langle 4, 1, R, 4\rangle$	
	$\langle 4, B, L, 5\rangle$	

States 5, 6 and 7 are as before. This time we test to see if the word is all ones. If not, the program jumps to 4 where it goes to the end, and to the routine from before. If it gets to the end without encountering a zero, it writes a one, returns to the beginning and deletes the initial symbol — so that the entire word is shifted one to the right. Then it goes to instruction 4 so that it goes to the right and works entirely as before. This time the output from (F) is,

		1	0	0	0		

as it should be. It is worthwhile to follow the actual operation of this and the previous program on one of the many Turing simulators available on the web (see E14.1).

More complex is a copy program to take an input x and return x.x. This program has four basic elements.

(1) A sort of control section which says what to do, depending on what sort of character we have in the original word.

(2) A program to copy 0; this will write a blank in the original word to "mark the spot"; move right to the second blank (across the blank between words, and to the blank to be filled); write a 0; move left to the original position, and replace the 0.

(3) Similarly a program to copy 1; this will write a blank in the original word to mark the spot; move right to the second blank; write a 1; move left to the original position, and replace the 1.

(4) And a program to move the head back to the original position when we are done.

Here is a program to do the job.

(H)

(1) *Control*	(2) *Copy* 0	(3) *Copy* 1
$\langle 1, 0, B, 10\rangle$	move from blank	move from blank
$\langle 1, 1, B, 20\rangle$	$\langle 10, B, R, 11\rangle$	$\langle 20, B, R, 21\rangle$
$\langle 1, B, L, 30\rangle$		
	right 2 blanks: 0	right 2 blanks: 1
(4) *Finish*	$\langle 11, 0, R, 11\rangle$	$\langle 21, 0, R, 21\rangle$
start of word	$\langle 11, 1, R, 11\rangle$	$\langle 21, 1, R, 21\rangle$
$\langle 30, 0, L, 30\rangle$	$\langle 11, B, R, 12\rangle$	$\langle 21, B, R, 22\rangle$
$\langle 30, 1, L, 30\rangle$		
$\langle 30, B, R, 0\rangle$	$\langle 12, 0, R, 12\rangle$	$\langle 22, 0, R, 22\rangle$
	$\langle 12, 1, R, 12\rangle$	$\langle 22, 1, R, 22\rangle$
	$\langle 12, B, 0, 13\rangle$	$\langle 22, B, 1, 23\rangle$
	left 2 blanks: 0	left 2 blanks: 1
	$\langle 13, 0, L, 13\rangle$	$\langle 23, 0, L, 23\rangle$
	$\langle 13, 1, L, 13\rangle$	$\langle 23, 1, L, 23\rangle$
	$\langle 13, B, L, 14\rangle$	$\langle 23, B, L, 24\rangle$
	$\langle 14, 0, L, 14\rangle$	$\langle 24, 0, L, 24\rangle$
	$\langle 14, 1, L, 14\rangle$	$\langle 24, 1, L, 24\rangle$
	$\langle 14, B, 0, 15\rangle$	$\langle 24, B, 1, 25\rangle$
	next char: return	next char: return
	$\langle 15, 0, R, 1\rangle$	$\langle 25, 1, R, 1\rangle$

You should be able to follow each stage.

E14.1. Study the copy program from the text along with the samples zero and suc from the course website. Then, starting with the file `blank.rb`, create Turing programs to compute the following. It will be best to submit your programs electronically.

a. copy(n). Takes input m and returns m.m. This is a simple implementation of the program from the text.

b. Create a Turing program to compute pred(n). Hint: Give your function two separate exit paths: One when the input is a string of 0s, returning with the input. In any other case, the output for input n is the predecessor of n. The method simply flips that for successor: From the right, change 0 to 1 until some 1 can be flipped to 0. There is no need to worry about the addition of a possible leading 0 to your result.

c. Create a Turing program to compute $\mathsf{ident}^3_3(\mathsf{x}, \mathsf{y}, \mathsf{z})$. For x.y.z observe that z might be longer than x and y put together; but, of course, it is not longer than x, y and z put together. Here is one way to proceed: Move to the start of the third word; use copy to generate x.y.z.z then plug spaces so that you have one long first word, xoyoz.z; you can mark the first position of the long word with a blank (and similarly, each time you write a character, mark the next position to the right with a blank so that you are always writing into the second blank up from the one where the character is read); then it is a simple matter of running a basic copy routine from right-to-left, and erasing junk when you are done.

14.1.2 Turing Computable Functions are Recursive

We turn now to showing that the (dextral) Turing computable functions are the same as the recursive functions. Our first aim is to show that every Turing computable function is recursive. But we begin with the simpler result that there is a recursive enumeration of Turing machines. We shall need this as we go forward, and it will let us compile some important preliminary results along the way.

The method is by now familiar. It will require some work, but we can do it in the same way as we approached recursive functions before. Begin by assigning to each symbol a *Gödel Number*.

a. $g[\mathsf{B}] = 3$	f. $g[\mathsf{L}] = 9$
b. $g[\mathsf{0}] = 5$	g. $g[\mathsf{R}] = 11$
c. $g[\mathsf{1}] = 7$	h. $g[\mathsf{q_i}] = 13 + 2i$

For a quadruple, say, $\langle \mathsf{q_1}, \mathsf{B}, \mathsf{L}, \mathsf{q_1} \rangle$, set $g = 2^{15} \times 3^3 \times 5^9 \times 7^{15}$. And for a sequence of quadruples with numbers $g_0, g_1 \ldots g_n$ the super Gödel number $g_s = 2^{g_0} \times 3^{g_1} \times \ldots \times \pi_n^{g_n}$. For convenience we frequently refer to the individual symbol codes with angle quotes around the symbol, so $\langle \mathsf{B} \rangle = 3$ where $\ulcorner \mathsf{B} \urcorner$, the number of the expression is 2^3.

Now we define a recursive function and some simple recursive relations,

$$\mathsf{lb}(\mathsf{v}) = 13 + 2\mathsf{v}$$

$$\textsc{lb}(\mathsf{n}) =_{\text{def}} (\exists \mathsf{v} \leq \mathsf{n})(\mathsf{n} = \mathsf{lb}(\mathsf{v}))$$

$$\textsc{sym}(\mathsf{n}) =_{\text{def}} \mathsf{n} = \langle \mathsf{B} \rangle \lor \mathsf{n} = \langle \mathsf{0} \rangle \lor \mathsf{n} = \langle \mathsf{1} \rangle$$

$$\textsc{act}(\mathsf{n}) =_{\text{def}} \mathsf{sym}(\mathsf{n}) \lor \mathsf{n} = \langle \mathsf{L} \rangle \lor \mathsf{n} = \langle \mathsf{R} \rangle$$

$$\textsc{quad}(\mathsf{n}) =_{\text{def}} \mathsf{len}(\mathsf{n}) = 4 \land \textsc{lb}(\mathsf{exp}(\mathsf{n}, 0)) \land \textsc{sym}(\mathsf{exp}(\mathsf{n}, 1)) \land \textsc{act}(\mathsf{exp}(\mathsf{n}, 2)) \land \textsc{lb}(\mathsf{exp}(\mathsf{n}, 3))$$

$\mathsf{lb}(\mathsf{v})$ is the Gödel number of instruction v. Then the relations are true when n is the number for an instruction label, a symbol, an action and a quadruple. In particular, a code for a quadruple numbers a sequence of four symbols of the appropriate sort.

We are now ready to number the Turing machines. For this, adopt a simple modification of our original specification: We have so-far supposed that a Turing machine might lack any given quadruple, say $\langle 3, 1, \mathsf{x}, \mathsf{y}\rangle$. In case it lacks this quadruple, if the machine reads 1 and is sent to state 3 it simply "hangs" with no place to go. Where q is the largest label in the machine, we now suppose that for any $\mathsf{p} \leq \mathsf{q}$, if no $\langle \mathsf{p}, \mathsf{C}, \mathsf{x}, \mathsf{y}\rangle$ is a member of the machine, the machine is simply supplemented with $\langle \mathsf{p}, \mathsf{C}, \mathsf{C}, \mathsf{p}\rangle$. The effect is as before: In this case, there is a place for the machine to go; but if the machine goes to $\langle \mathsf{p}, \mathsf{C}, \mathsf{C}, \mathsf{p}\rangle$, it remains in that state, repeating it over and over. In the case of label 0, the states are added to the machine, but serve no function, as the zero label forces halt. Further, we suppose that the quadruples in a Turing machine are taken in order, $\langle 0, 0, \mathsf{x}, \mathsf{y}\rangle$, $\langle 0, 1, \mathsf{x}, \mathsf{y}\rangle$, $\langle 0, \mathsf{B}, \mathsf{x}, \mathsf{y}\rangle$, $\langle 1, 0, \mathsf{x}, \mathsf{y}\rangle \ldots \langle \mathsf{q}, 0, \mathsf{x}, \mathsf{y}\rangle$, $\langle \mathsf{q}, 1, \mathsf{x}, \mathsf{y}\rangle$, $\langle \mathsf{q}, \mathsf{B}, \mathsf{x}, \mathsf{y}\rangle$. So each Turing machine has a unique specification. On this account, a Turing machine halts only when it reaches a state of the sort $\langle \mathsf{x}, \mathsf{x}, \mathsf{x}, 0\rangle$. And the ordered specification itself guarantees the functional requirement – that there are no two quadruples with the first inputs the same and the latter different. So for $\textsc{tmach}(\mathsf{n})$,

$$(\exists \mathsf{w} < \mathsf{len}(\mathsf{n}))(\mathsf{len}(\mathsf{n}) = 3 \times (\mathsf{w} + 2)) \wedge (\forall \mathsf{v}, 3 \times \mathsf{v} + 2 < \mathsf{len}(\mathsf{n}))(\forall \mathsf{x} \leq \mathsf{n})\{$$
$$[\mathsf{x} = \mathsf{exp}(\mathsf{n}, 3 \times \mathsf{v}) \rightarrow (\textsc{quad}(\mathsf{x}) \wedge \mathsf{exp}(\mathsf{x}, 0) = \mathsf{lb}(\mathsf{v}) \wedge \mathsf{exp}(\mathsf{x}, 1) = \langle 0\rangle)] \wedge$$
$$[\mathsf{x} = \mathsf{exp}(\mathsf{n}, 3 \times \mathsf{v} + 1) \rightarrow (\textsc{quad}(\mathsf{x}) \wedge \mathsf{exp}(\mathsf{x}, 0) = \mathsf{lb}(\mathsf{v}) \wedge \mathsf{exp}(\mathsf{x}, 1) = \langle 1\rangle)] \wedge$$
$$[\mathsf{x} = \mathsf{exp}(\mathsf{n}, 3 \times \mathsf{v} + 2) \rightarrow (\textsc{quad}(\mathsf{x}) \wedge \mathsf{exp}(\mathsf{x}, 0) = \mathsf{lb}(\mathsf{v}) \wedge \mathsf{exp}(\mathsf{x}, 1) = \langle \mathsf{B}\rangle)]\}$$

Given our modifications, the length of a Turing machine must be a non-zero multiple of three including at least the initial labels zero and one. So for some w, $\mathsf{len}(\mathsf{n}) = 3 \times (\mathsf{w} + 2)$. Then for each initial label v, there are three quadruples; so there are quadruples $3 \times \mathsf{v}$, $3 \times \mathsf{v} + 1$ and $3 \times \mathsf{v} + 2$, taken in the standard order, and each with initial label v. Since n is a super Gödel number, and each x the number of a quadruple it is the exponents of x that reveal the instruction label and cell content.

But now it is easy to see,

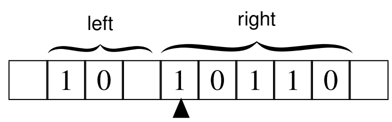
T14.1. There is a recursive enumeration of the Turing machines. Set,

$$\mathsf{mach}(0) = \mu \mathsf{z}[\textsc{tmach}(\mathsf{z})]$$
$$\mathsf{mach}(\mathsf{Sn}) = \mu \mathsf{z}[\mathsf{z} > \mathsf{mach}(\mathsf{n}) \wedge \textsc{tmach}(\mathsf{z})]$$

Since $\mathsf{mach}(\mathsf{n})$ is is a recursive function from the natural numbers onto the Turing machines, they are recursively enumerable. While this enumeration is recursive, it is not primitive recursive.

Now, as we work toward a demonstration that Turing computable functions are recursive, let us pause for some key ideas. Consider a tape divided as follows,

(I)



We shall code the tape with a pair of numbers. Where at any stage the head divides the tape into left and right parts, first a standard code for the right hand side, $\ulcorner 10110 \urcorner$, and second, a code for the left side read from the inside out $\ulcorner B01 \urcorner$. Taken as a pair, these numbers record at once contents of the tape, and the position of the head — always under the first digit of the coded right number.

Say a dextral Turing machine computes a function $\mathsf{f(n)} = \mathsf{m}$. Let us suppose that we have functions $\mathsf{code(n)}$ and $\mathsf{decode(m)}$ to move between m and n and their codes (where this requires moving from the numbers m and n through their binary representations, and then to the codes). So we concentrate on the machine itself, and wish to track the status of the Turing machine i given input n for each step j of its operation. In order to track the status of the machine, we shall require functions $\mathsf{left(i, n, j)}$, $\mathsf{right(i, n, j)}$ to record codes of the left and right portions of the tape, and $\mathsf{state(i, n, j)}$ for the current quadruple state of the machine.

First, as we have observed, for any Turing machine, there is a unique quadruple for any instruction label and tape value. Thus, $\mathsf{machs(i, m, n)}$ numbers a quadruple as a function of the number of the machine in the enumeration, and Gödel numbers for initial label and tape value. Thus $\mathsf{machs(i, m, n)}$ is,

$$(\mu \mathsf{y} \leq \mathsf{mach(i)})(\exists \mathsf{v} < \mathsf{len(mach(i))})[\mathsf{y} = \mathsf{exp(mach(i), v)} \wedge \mathsf{exp(y, 0)} = \mathsf{m} \wedge \mathsf{exp(y, 1)} = \mathsf{n}]$$

So $\mathsf{machs(i, m, n)}$ returns the number of that quadruple in machine i whose initial label has number m, and initial value number n. Since the machine is a function, there must be a unique state with those initial values.

In addition, where $\mathsf{n} = \mathsf{a} \star \mathsf{b}$, let us adopt a sort of converse to concatenation such that $\mathsf{a} \circ \mathsf{n} = \mathsf{b}$.

$$\mathsf{a} \circ \mathsf{n} = (\mu \mathsf{x} \leq \mathsf{n})(\forall \mathsf{i} < \mathsf{len(n)} \dot{-} \mathsf{len(a)})(\mathsf{exp(x, i)} = \mathsf{exp(n, len(a)} + \mathsf{i}))$$

So we want the least x such that its length is the length of n less the length of a, and the values of x at any position i are the same as those of n at $\mathsf{len(a)} + \mathsf{i}$. Thus $\mathsf{a} \circ \mathsf{n}$ "lops off" the portion numbered a from the expression numbered n.

Recall that our Turing machine is to calculate a function $\mathsf{f(n) = m}$. Initial values of $\mathsf{left(i,n,j)}$, $\mathsf{right(i,n,j)}$ and $\mathsf{state(i,n,j)}$ are straightforward.

$$\mathsf{left(i,n,0)} = \ulcorner \mathsf{BB} \urcorner$$

$$\mathsf{right(i,n,0)} = \mathsf{code(n)}$$

$$\mathsf{state(i,n,0)} = \mathsf{machs(i, \langle 1 \rangle, exp(right(i,n,0),0))}$$

On a dextral machine, the machine never writes to the left of its initial position, and the head never moves more than one position to the left of its initial position; so we simply set the value of the left portion to a couple of blanks. This ensures that there is enough "space" on the left for the machine to operate (and that, for any position of the machine head, there is always a left portion of the tape). The starting right number is just the code of the input to the function. And the initial state value is determined by the input label $\mathsf{1}$ and the first value on the tape which is coded by the first exponent of $\mathsf{right(i,n,0)}$.

For the successor values,

$$\mathsf{left(i,n,Sj)} = \begin{cases} \mathsf{left(i,n,j)} & \text{if} \quad \textsc{sym}\mathsf{(exp(state(i,n,j),2))} \\ \mathsf{2^{exp(right(i,n,j),0)} \star left(i,n,j)} & \text{if} \quad \mathsf{exp(state(i,n,j),2) = \langle R \rangle} \\ \mathsf{2^{exp(left(i,n,j),0)} \circ left(i,n,j)} & \text{if} \quad \mathsf{exp(state(i,n,j),2) = \langle L \rangle} \end{cases}$$

If a symbol is written in the current cell, there is no change in the left number. If the head moves to the left or the right, the first value is either appended or deleted, depending on direction. And similarly for $\mathsf{right(i,n,Sj)}$ but with separate clauses for each of the symbols that may be written onto the first position. And now the successor value for state is determined by the Turing machine together with the new label and the value under the head after the current action has been performed.

$$\mathsf{state(i,n,Sj)} = \mathsf{machs(i, exp(state(i,n,j),3), exp(right(i,n,Sj),0))}$$

The machine jumps to a new state depending on the label and value on the tape. Observe that we are here proceeding by *simultaneous* recursion, defining multiple functions together. It should be clear enough how this works (see E12.26, p. 587).

If the machine enters a zero state then it halts. So set,

$$\mathsf{stop(i,n,j)} =_{\text{def}} (\mu \mathsf{y} \leq \mathsf{len(mach(i))})(\mathsf{exp(state(i,n,j),0) = lb(y)})$$

$\mathsf{exp(state(i,n,j),0)}$ is the number of of the instruction label. So $\mathsf{exp(state(i,n,j),0)} = \mathsf{lb(y)}$ when y is the label. And $\mathsf{stop(i,n,j)}$ takes the value $\mathsf{0}$ just in case machine i with input n is halted at step j. When the first member of $\mathsf{state(i,n,j)}$ codes zero, the machine is halted, otherwise it is running. So y takes the value zero just in case the machine is halted.

T14.2. Every Turing computable function is a recursive function. Supposing Turing machine i computes a function $\mathsf{f(n)}$,

$$\mathsf{f(n) = decode(right(i, n, \mu j[stop(i, n, j) = 0]))}$$

When a dextral Turing machine stops, the value of right is just the code of its output value m; so if we decode $\mathsf{right(i, n, j)}$ at that stage, we have the value of the function calculated by the Turing machine. Supposing, as we have that the machine does return a value, minimization operates on a regular function. Since this function is recursive, the function calculated by Turing machine i is a recursive function.

E14.2. Find a recursive function to calculate $\mathsf{right(i, n, j)}$. Hint: You might find a combination of $\star$ and $\circ$ useful for the case when a symbol is written into the first cell.

E14.3. Find a recursive function to calculate $\mathsf{decode(n)}$.

E14.4. Suppose a "dual" Turing machine has two tapes, with a machine head for each. Instructions are of the sort $\langle \mathsf{q_i, C_{t_a}, A_{t_b}, q_j} \rangle$ where $\mathsf{t_a}$ and $\mathsf{t_b}$ indicate the relevant tape. Show that every function that is dual Turing computable is recursive.

14.1.3 Recursive Functions are Turing Computable

To show that the recursive functions are identical to the Turing computable functions, we now show that all recursive functions are Turing computable.

T14.3. Every recursive function is Turing computable.

Suppose $\mathsf{f(\vec{x})}$ is a recursive function. Then there is a sequence of recursive functions $\mathsf{f_0}$, $\mathsf{f_1 \dots f_n}$ such that $\mathsf{f_n = f}$, where each member is either an initial function or arises from previous members by composition, recursion, or regular minimization. The argument is by induction on this sequence.

Basis: We have already seen that the initial functions $\mathsf{zero(x)}$, $\mathsf{suc(x)}$ and $\mathsf{idnt^j_k}$, as illustrated in E14.1, are Turing computable.

Assp: For any i, $0 \leq i < k$, $\mathsf{f_i(\vec{x})}$ is Turing computable.

Show: $\mathsf{f_k}(\vec{\mathsf{x}})$ is Turing computable.

$\mathsf{f_k}$ is either an initial function or arises from previous members by composition, recursion, or regular minimization. If an initial function, then as in the basis. So suppose $\mathsf{f_k}$ arises from previous members.

(c) $\mathsf{f_k}(\vec{\mathsf{x}}, \vec{\mathsf{y}}, \vec{\mathsf{z}})$ arises by composition from $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$. By assumption $\mathsf{g}(\vec{\mathsf{y}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{w}, \vec{\mathsf{z}})$ are Turing computable. For the simplest case, consider $\mathsf{h}(\mathsf{g}(\mathsf{y}))$: Chain together Turing programs to calculate $\mathsf{g}(\mathsf{y})$ and then $\mathsf{h}(\mathsf{w})$ — so the first program operates upon y to calculate $\mathsf{g}(\mathsf{y})$ and the second begins where the first leaves off, operating on the result to calculate $\mathsf{h}(\mathsf{g}(\mathsf{y}))$. A case like $\mathsf{h}(\mathsf{x}, \mathsf{g}(\mathsf{y}), \mathsf{z})$ is more complex insofar as $\mathsf{g}(\mathsf{y})$ may take up a different number of cells from y: it is sufficient to run a copy to get $\mathsf{x.y.z.y}$; then $\mathsf{g}(\mathsf{y})$ to get $\mathsf{x.y.z.g(y)}$; then copy for $\mathsf{x.y.z.g(y).z}$ and a copy that replaces the last two numbers to get $\mathsf{x.g(y).z}$. Then you can run h. And similarly in other cases.

(r) $\mathsf{f_k}(\vec{\mathsf{x}}, \mathsf{y})$ arises by recursion from $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$. By assumption $\mathsf{g}(\vec{\mathsf{x}})$ and $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ are Turing computable. Recall our little programs from chapter 12 which begin by using $\mathsf{g}(\vec{\mathsf{x}})$ to find $\mathsf{f}(0)$ and then use $\mathsf{h}(\vec{\mathsf{x}}, \mathsf{y}, \mathsf{u})$ repeatedly for y in 0 to $\mathsf{b}-1$ to find the value of $\mathsf{f}(\vec{\mathsf{x}}, \mathsf{b})$ (see, for example, p. 548). For a representative case, consider $\mathsf{f}(\mathsf{m}, \mathsf{b})$.

a. Produce a sequence,
$$\mathsf{m.b.m.b-1.m.b-2\ldots m.2.m.1.m.0.m}$$
This requires a $\mathsf{copypair}(\mathsf{x}, \mathsf{y})$ that takes $\mathsf{m.n}$ and returns $\mathsf{m.n.m.n}$ and $\mathsf{pred}(\mathsf{x})$. Given $\mathsf{m.b}$ on the tape, run $\mathsf{copypair}$ to get $\mathsf{m.b.m.b}$ (and mark the first m with a blank). Then loop as follows: run pred on the final b; if it is already 0, erase final 0, go to the previous m and move on to (b); otherwise, move to previous m, run $\mathsf{copypair}$ and loop.

b. Run g on the last block of digits m. This gives,
$$\mathsf{m.b.m.b-1.m.b-2\ldots m.2.m.1.m.0.f(m,0)}$$

c. Back up to the previous m and run h on the concluding three blocks $\mathsf{m.0.f(m,0)}$. This gives,
$$\mathsf{m.b.m.b-1.m.b-2\ldots m.2.m.1.f(m,1)}$$
And so forth. Stop when you reach the m with an extra blank (with two blanks in a row). At that stage, we have, $\mathsf{m^*.b.f(m,b)}$. Fill the first blank, run idnt^3_3 and you are done. Observe that the original $\mathsf{m.b}$ plays no role in the calculation other to serve as the initial template for the series, and then as an end marker on your way back up — there is never

a need to apply h to any value greater than $\mathsf{b}-1$ in the calculation of $\mathsf{f(m,b)}$.

(m) $\mathsf{f_k}(\vec{\mathsf{x}})$ arises by regular minimization from $\mathsf{g}(\vec{\mathsf{x}},\mathsf{y})$. By assumption, $\mathsf{g}(\vec{\mathsf{x}},\mathsf{y})$ is Turing computable. For a representative case, suppose we are given m and want $\mu\mathsf{y}[\mathsf{g(m,y)}=0]$.

a. Given, m, produce $\mathsf{m.0.m.0}$.

b. From a tape of the form $\mathsf{m.y.m.y}$ loop as follows: Move to the second m; run g on $\mathsf{m.y}$; this gives $\mathsf{m.y.g(m,y)}$; check to see if the result is zero; if it is, run idnt^3_2 and you are done (this is the same as deleting the last zero and running idnt^2_2); if the result is not zero, delete $\mathsf{g(m,y)}$ to get $\mathsf{m.y}$; run suc on y; and then a copier to get $\mathsf{m.y'.m.y'}$, and loop. The loop halts when it reaches the value of y for which g has output 0 — and there must be some such value if g is regular.

Indct: Any recursive function $\mathsf{f}(\vec{\mathsf{x}})$ is Turing computable.

And from T14.2 together with T14.3, the Turing computable functions are identical to the recursive functions. It is perhaps an "amazing" coincidence — that functions independently defined in these ways should turn out to be identical. And we have here the beginnings of an idea behind Church's thesis which we shall explore in section 14.3.

E14.5. From exercise E14.1 you should already have Turing programs for $\mathsf{suc(x)}$, $\mathsf{pred(x)}$, $\mathsf{copy(x)}$ and $\mathsf{idnt}^3_3\mathsf{(x,y,z)}$. Now produce each of the following, in order, leading up to the recursive addition function. When you require one as part of another simply copy it into the larger file.

a. The function, $\mathsf{h(x,y,u)}$. For addition, $\mathsf{h(x,y,u)}$ is $\mathsf{suc(idnt}^3_3\mathsf{(x,y,u))}$. So this is a simple combination of suc and idnt^3_3. For addition, $\mathsf{g(x)}=\mathsf{idnt}^1_1\mathsf{(x)}=\mathsf{x}$, which requires no action; so we will not worry about that.

b. The function, $\mathsf{copypair}$. Take $\mathsf{a.b}$ and return $\mathsf{a.b.a.b}$. One approach is to produce a simple modification of copy that takes $\mathsf{a.b}$ and produces $\mathsf{a.b.a}$. Run this program starting at a, and then another copy of it starting at b.

c. The function, $\mathsf{cascade}$. This is the program to produce $\mathsf{m.n.m.n-1.m.n-2}$ $\ldots\mathsf{m.0.m}$. The key elements are $\mathsf{copypair}$ and pred. The main loop runs pred on the last word; if the object is not zero, back up one and run $\mathsf{copypair}$; and

so forth. To prepare for the next stage, you should begin by running copypair and then "damage" the very first m by putting a blank in its first cell. Let the program finish with the head on m at the end.

d. The function, $\mathsf{plus}(\mathsf{m}, \mathsf{n})$. From m at the far right of the sequence, back up two words; check to see if there is an extra blank; if so, run idnt^3_3 and you are done; if not, run $\mathsf{h}(\mathsf{x}, \mathsf{y}, \mathsf{u})$. Though m.n is part of the "cascade" series, we never run h on m.n.u. In a program we may make use of m.n as described, but in damaged form — as an end marker for the series.

   There are easier ways to do addition on a Turing machine! The obvious strategy is to put m in a location x and n in a location y; run pred on the value in location x and then suc on the value in location y; the result appears in y when pred hits zero. The advantage of our approach is that it illustrates (an important case of) the demonstration that a Turing machine can compute any recursive function.

E14.6. Produce each of the following, leading up to a Turing program for the function $\mu\mathsf{y}[\mathsf{ch}(\mathsf{x} = \mathsf{pred}(\mathsf{y})) = 0]$, that is the function which returns the least y such that x equals the predecessor of y — such that the characteristic function of $\mathsf{x} = \mathsf{pred}(\mathsf{y})$ returns 0.

a. The function $\mathsf{idnt}^2_2(\mathsf{x}, \mathsf{y})$. This can be a simple modification of idnt^3_3.

b. The function $\mathsf{ch}(\mathsf{x} = \mathsf{y})$, which returns 0 when $\mathsf{x} = \mathsf{y}$ and otherwise 1. This is, of course, a recursive function. But you can get it more efficiently and more directly. To compare numbers, you have to worry about leading zeros that might make equivalent numbers physically distinct. Here is one strategy: From x.y check to see if one or both are all zeros; exit with 1 or 0 in the different cases; if neither works, apply pred to x and to y and return to the start; eventually you will come to a stage where the check for zero returns a result.

c. The function $\mathsf{ch}(\mathsf{x} = \mathsf{pred}(\mathsf{y}))$. This is a simple case of composition.

d. The function $\mu\mathsf{y}[\mathsf{ch}(\mathsf{x} = \mathsf{pred}(\mathsf{y})) = 0]$, by the routine discussed in the text.

   Of course, for any number except 0, this is nothing but a long-winded equivalent to $\mathsf{suc}(\mathsf{x})$. The point, however, is to apply the algorithm for regular minimization, and so to work through the last stage of the demonstration that recursive functions are Turing computable.

14.2 Essential Results

In chapter 12 essential results were built on the diagonal lemma (T12.19). This time, we depend on a *halting problem* with special application to Turing machines. Once we have established the halting problem, results like ones from before follow in short order.

14.2.1 Halting

A Turing machine is a set of quadruples. Things are arranged so that Turing machines do not "hang" in the sense that they reach a state with no applicable instruction. But a Turing machine may go into a loop or routine from which it never emerges. That is, a Turing machine may or may not *halt* in a finite number of steps. This raises the question whether there is a general way to *tell* whether Turing machines halt when started on a given input. This is an issue of significance for computing theory. And, as we shall see, the answer has consequences beyond computing.

The problem divides into narrower "self-halting" and broader "general halting" versions. First, the self-halting problem: By T14.1 there is an enumeration of the Turing machines. Consider some such enumeration, Π_0, $\Pi_1 \ldots$ of Turing machines and an array as follows,

$$\text{(J)}\qquad \begin{array}{c|cccc} & 0 & 1 & 2 & \ldots \\ \hline \Pi_0 & \boldsymbol{\Pi}_0(0) & \Pi_0(1) & \Pi_0(2) & \\ \Pi_1 & \Pi_1(0) & \boldsymbol{\Pi}_1(1) & \Pi_1(2) & \\ \Pi_2 & \Pi_2(0) & \Pi_2(1) & \boldsymbol{\Pi}_2(2) & \\ \vdots & & & & \end{array}$$

We run Π_0 on inputs $0, 1 \ldots$; Π_1 on $0, 1 \ldots$; and so forth. Now ask whether there is a Turing program (that is, a recursive function) to decide in general whether Π_i halts when applied to its own number in the enumeration — a program $\mathsf{H}(i)$ such that $\mathsf{H}(i) = 0$ if $\Pi_i(i)$ halts, and $\mathsf{H}(i) = 1$ if $\Pi_i(i)$ does not halt.

T14.4. There is no Turing machine $\mathsf{H}(i)$ such that $\mathsf{H}(i) = 0$ if $\Pi_i(i)$ halts and $\mathsf{H}(i) = 1$ if it does not.

Suppose otherwise. That is, suppose there is a halting machine $\mathsf{H}(i)$ where for any $\Pi_i(i)$, $\mathsf{H}(i) = 0$ if $\Pi_i(i)$ halts and $\mathsf{H}(i) = 1$ if it does not. Chain this program into a simple looping machine $\Lambda(j)$ defined as follows,

$\langle q, 0, 0, q\rangle$
$\langle q, 1, 1, 0\rangle$

So when $\mathsf{j} = 0$, Λ goes into an infinite loop, remaining in state q forever; when $\mathsf{j} = 1$, Λ halts gracefully with output 1. Let the combination of H and Λ be $\Delta(\mathsf{i})$; so $\Delta(\mathsf{i})$ calculates $\Lambda(\mathsf{H}(\mathsf{i}))$. On our assumption that there is a Turing machine $\mathsf{H}(\mathsf{i})$, the machine Δ must appear in the enumeration of Turing machines with some number d.

But this is impossible. Consider $\Delta(\mathsf{d})$ and suppose $\Delta(\mathsf{d})$ halts; since Δ halts on input d, the halting machine, $\mathsf{H}(\mathsf{d}) = 0$; and with this input, Λ goes into the infinite loop; so the composition $\Lambda(\mathsf{H}(\mathsf{d}))$ does not halt; and this is just to say $\Delta(\mathsf{d})$ does not halt. Reject the assumption, $\Delta(\mathsf{d})$ does not halt. But since $\Delta(\mathsf{d})$ does not halt, the halting machine $\mathsf{H}(\mathsf{d}) = 1$; and with this input, Λ halts gracefully with output 1; so the composition $\Lambda(\mathsf{H}(\mathsf{d}))$ halts; and this is just to say $\Delta(\mathsf{d})$ halts. Reject the original assumption, there is no machine $\mathsf{H}(\mathsf{i})$ which says whether an arbitrary $\Pi_{\mathsf{i}}(\mathsf{i})$ halts.

For this argument, it is important that H is a component of Δ. Information about whether Δ halts gives information about the behavior of H, and information about the behavior of H, about whether Δ halts.

The more general question is whether there is a machine to decide for any Π_{i} and n whether $\Pi_{\mathsf{i}}(\mathsf{n})$ halts. But it is immediate that if there is no Turing machine to decide the more narrow self-halting problem, there is no Turing machine to decide this more general version.

T14.5. There is no Turing machine $\mathsf{H}(\mathsf{i},\mathsf{n})$ such that $\mathsf{H}(\mathsf{i},\mathsf{n}) = 0$ if $\Pi_{\mathsf{i}}(\mathsf{n})$ halts and $\mathsf{H}(\mathsf{i},\mathsf{n}) = 1$ if it does not.

Suppose otherwise. That is, suppose there is a halting machine $\mathsf{H}(\mathsf{i},\mathsf{n})$ where for any $\Pi_{\mathsf{i}}(\mathsf{n})$, $\mathsf{H}(\mathsf{i},\mathsf{n}) = 0$ if $\Pi_{\mathsf{i}}(\mathsf{n})$ halts and $\mathsf{H}(\mathsf{i},\mathsf{n}) = 1$ if it does not. Chain this program after a copier $\mathsf{K}(\mathsf{n})$ which takes input n and gives $\mathsf{n}.\mathsf{n}$. The combination $\mathsf{H}(\mathsf{K}(\mathsf{i}))$ decides whether $\Pi_{\mathsf{i}}(\mathsf{i})$ halts. This is impossible; reject the assumption: There is no such Turing machine $\mathsf{H}(\mathsf{i},\mathsf{n})$.

And when combined with T14.3 according to which every recursive function is Turing computable, these theorems which tell us that no Turing program is sufficient to solve the halting problem, yield the result that no recursive function solves the halting problem: if a function is recursive, then it is Turing computable; and since it is Turing computable, it does not solve the halting problem. Observe that we may be able to decide in particular cases whether a program halts. No doubt you have been able to do so in particular cases! What we have shown is that there is no perfectly general recursive method to decide whether $\Pi_{\mathsf{i}}(\mathsf{n})$ halts.

E14.7. Consider again the μ-recursive functions introduced in E12.7. Suppose that these functions can be numbered and that there is a μ-recursive function $\mathsf{emr(i)}$ to enumerate them; so $\mathsf{emr(i)}$ returns the Gödel number of the i^{th} function in the enumeration. (You will have occasion to produce this function in a later exercise.) Show that there is no μ-recursive function $\mathsf{def(i)}$ such that $\mathsf{def(i)} = 0$ if $\mathsf{f_i(i)}$ is defined and $\mathsf{def(i)} = 1$ if $\mathsf{f_i(i)}$ is undefined. Hint: Let your diagonal function $\mathsf{diag(i)} = \mu\mathsf{y}[\mathsf{def(i)} = \mathsf{y} \wedge \mathsf{y} = 1]$. We might think of this as the *definition problem*.

14.2.2 The Decision Problem

Recall our demonstration from section 12.5.2 that if Q is consistent then no recursive relation identifies the theorems of predicate logic. With the identity between the recursive functions and the Turing computable functions, this is the same as the result that if Q is consistent then no Turing computable function identifies the theorems of predicate logic. We are now in a position to obtain a related result directly, by means of the halting problem. Recall from chapter 13 (p. 612) that a theory T is *ω-inconsistent* iff for some $\mathcal{P}(x)$, T proves each $\mathcal{P}(\overline{\mathsf{m}})$ but also proves $\sim\forall x\mathcal{P}(x)$. Equivalently, T is ω-inconsistent iff for every m, can prove each $T \vdash \sim\mathcal{P}(\overline{\mathsf{m}})$ and $T \vdash \exists x\mathcal{P}(x)$. We show,

T14.6. If Q is ω-consistent, then no Turing computable function $\mathsf{f(n)}$ is such that $\mathsf{f(n)} = 0$ just in case n numbers a theorem of predicate logic.

Suppose Q is ω-consistent, and consider our recursive function $\mathsf{stop(i, n, j)}$ which takes the value 0 iff $\Pi_{\mathsf{i}}(\mathsf{n})$ is halted. Since it is recursive, stop is captured by some $Stop(i, n, j, z)$ so that,

(i) If $\Pi_{\mathsf{i}}(\mathsf{i})$ is halted by step j, Q $\vdash Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$

(ii) If $\Pi_{\mathsf{i}}(\mathsf{i})$ never halts, Q $\vdash \sim Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$ for any j

For any i, let $\mathcal{H}(\overline{\mathsf{i}}) = \exists z Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, z, \emptyset)$. Then if $\Pi_{\mathsf{i}}(\mathsf{i})$ halts, there is some j such that Q $\vdash Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$; so Q $\vdash \mathcal{H}(\overline{\mathsf{i}})$. And if $\Pi_{\mathsf{i}}(\mathsf{i})$ never halts, for every j, Q $\vdash \sim Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{j}}, \emptyset)$; so since Q is ω-consistent, Q $\nvdash \mathcal{H}(\overline{\mathsf{i}})$. So where $\mathcal{Q}$ is a conjunction of the axioms of Q, if $\Pi_{\mathsf{i}}(\mathsf{i})$ halts $\vdash \mathcal{Q} \to \mathcal{H}(\overline{\mathsf{i}})$ and if $\Pi_{\mathsf{i}}(\mathsf{i})$ never halts $\nvdash \mathcal{Q} \to \mathcal{H}(\overline{\mathsf{i}})$; so,

$$\vdash \mathcal{Q} \to \mathcal{H}(\overline{\mathsf{i}}) \quad \text{iff} \quad \Pi_{\mathsf{i}}(\mathsf{i}) \text{ halts}$$

Suppose some Turing computable function $\mathsf{f(n)}$ takes the value 0 just in case n numbers a theorem. Then for any i, f applied to $\ulcorner\mathcal{Q} \to \mathcal{H}(\overline{\mathsf{i}})\urcorner$ takes the value

0 iff $\mathcal{Q} \rightarrow \mathcal{H}(\overline{\mathsf{i}})$ is a theorem, iff $\Pi_{\mathsf{i}}(\mathsf{i})$ halts. But this is impossible; reject the assumption: If Q is ω-consistent, then there is no Turing computable function that returns the value zero for numbers of theorems of predicate logic.

And, of course, this result according to which if Q is ω-consistent no Turing computable function returns zero just for theorems of predicate logic is equivalent to the result that if Q is ω-consistent, then no recursive function returns zero just for theorems of predicate logic.[3]

E14.8. Return again to the μ-recursive functions from the previous exercise (with E12.7 and E12.16). Suppose that addition to $\mathsf{enf}(\mathsf{i})$ to enumerate the functions there is a μ-recursive $\mathsf{val}(\mathsf{i},\mathsf{n})$ to return the value of $\mathsf{f}_{\mathsf{i}}(\mathsf{n})$; so $\mathsf{val}(\mathsf{i},\mathsf{n}) = \mathsf{f}_{\mathsf{i}}(\mathsf{n})$. (Again, you will have the opportunity to construct this function in a later exercise.) From E12.16 this function is captured in $\mathrm{Q_s}$ by some $Val(i,n,y)$. Use your result from the definition problem in E14.7 to show that if $\mathrm{Q_s}$ is ω-consistent, then no μ recursive function $\mathsf{f}(\mathsf{n})$ is such that $\mathsf{f}(\mathsf{n}) = 0$ just in case n numbers a theorem of predicate logic. Hint: Let $Def(\overline{\mathsf{i}}) =_{\text{def}} \exists z Val(\overline{\mathsf{i}},\overline{\mathsf{i}},z)$.

14.2.3 Incompleteness Again

In T12.21 we saw that no consistent, recursively axiomatizable theory extending Q is negation complete. We shall see this again. However, as described in chapter 13, the incompleteness result comes in different forms. In particular, the one as from chapter 12 which depends on consistency and capture, and another which depends on soundness and expression. We are positioned to see the result in both forms.

Semantic Version

A key preliminary to the chapter 12 demonstration of incompleteness is T12.20 which applies the diagonal lemma to show that for no consistent theory T extending Q is a recursive relation true of (numbers for) its theorems. This time, by means of the halting result, we show that the *truths* of $\mathcal{L}_{\mathrm{NT}}$ are not recursively enumerable.

T14.7. The set of truths of $\mathcal{L}_{\mathrm{NT}}$ is not recursively enumerable.

[3] This argument, and the parallel one in chapter 12 have the advantage of simplicity. However, this result that no recursive function is true just of the theorems of predicate logic need not be conditional on the consistency (or ω-consistency) of Q. For an illuminating version of the strengthened result from the halting problem, see chapter 11 of Boolos et al., *Computability and Logic*.

Consider again our recursive function $\mathsf{stop}(\mathsf{i}, \mathsf{n}, \mathsf{j})$; since it is recursive, it is expressed by some $Stop(i, n, j, z)$; for arbitrary i, set $\mathcal{H}(\overline{\mathsf{i}}) = \exists z Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, z, \emptyset)$. Suppose some Π_{e} enumerates the truths of $\mathcal{L}_{NT}$, halting with output $\mathsf{0}$ if (the number for) $\mathcal{H}(\overline{\mathsf{i}})$ appears in the enumeration, and with output $\mathsf{1}$ if $\sim\mathcal{H}(\overline{\mathsf{i}})$ appears. Exactly one of $\mathcal{H}(\overline{\mathsf{i}})$ or $\sim\mathcal{H}(\overline{\mathsf{i}})$ is true; so the number for one of them will eventually turn up since Π_{e} enumerates all the truths of $\mathcal{L}_{NT}$.

(i) Suppose $\mathsf{N}[\mathcal{H}(\overline{\mathsf{i}})] = \mathsf{T}$; then for some m, $\mathsf{N}[Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{m}}, \emptyset)] = \mathsf{T}$; so $\mathsf{N}[\sim Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{m}}, \emptyset)] \neq \mathsf{T}$; so by expression, $\langle\langle \mathsf{i}, \mathsf{i}, \mathsf{m}\rangle, \emptyset\rangle \in \mathsf{stop}$; so $\Pi_{\mathsf{i}}(\mathsf{i})$ stops.

(ii) Suppose $\mathsf{N}[\sim\mathcal{H}(\overline{\mathsf{i}})] = \mathsf{T}$; then $\mathsf{N}[\mathcal{H}(\overline{\mathsf{i}})] \neq \mathsf{T}$; so for any $\mathsf{m} \in \mathsf{U}$, $\mathsf{N}[Stop(\overline{\mathsf{i}}, \overline{\mathsf{i}}, \overline{\mathsf{m}}, \emptyset)] \neq \mathsf{T}$; so by expression, $\langle\langle \mathsf{i}, \mathsf{i}, \mathsf{m}\rangle, \emptyset\rangle \notin \mathsf{stop}$; so $\Pi_{\mathsf{i}}(\mathsf{i})$ never stops.

So Π_{e}, halts with output $\mathsf{0}$ iff $\mathsf{N}[\mathcal{H}(\overline{\mathsf{i}})] = \mathsf{T}$ (by its definition); iff $\Pi_{\mathsf{i}}(\mathsf{i})$ halts (by (i) and (ii)); so Π_{e} solves the halting problem. This is impossible; there is no such Turing machine. And since no Turing machine enumerates the truths of $\mathcal{L}_{NT}$, no recursive function enumerates the truths of $\mathcal{L}_{NT}$.

This theorem, together with T12.17 which tells us that if T is a recursively axiomatized formal theory then the set of theorems of T is recursively enumerable, puts us in a position to obtain an incompleteness result mirroring T13.2.

T14.8. If T is a recursively axiomatized sound theory whose language includes $\mathcal{L}_{NT}$, then T is negation incomplete.

Suppose T is a recursively axiomatized sound theory whose language includes $\mathcal{L}_{NT}$. By T12.17, there is an enumeration of the theorems of T, and since T is sound, all of the theorems in the enumeration are true. But by T14.7, there is no enumeration of all the truths of $\mathcal{L}_{NT}$; so the enumeration of theorems is not an enumeration of all truths; so some true $\mathcal{P}$ is not among the theorems of of T; and since $\mathcal{P}$ is true, $\sim\mathcal{P}$ is not true; and since T is sound, neither is $\sim\mathcal{P}$ among the theorems of T. So $T \nvdash \mathcal{P}$ and $T \nvdash \sim\mathcal{P}$.

This incompleteness result requires the *soundness* of T, where where soundness is more than mere consistency. But it requires only that the language include $\mathcal{L}_{NT}$ and so have the power to *express* recursive functions — where this leaves to the side a requirement that T extends Q, and so be able to capture recursive functions.

Syntactic Version

From the halting problem, we can obtain the other sort of incompleteness result as well. Thus we have a theorem like the combination of T13.4 and T13.5.

T14.9. If T is a recursively axiomatized theory extending Q, then there is a sentence $\mathscr{P}$ such that if T is consistent $T \nvdash \mathscr{P}$, and if T is ω-consistent, $T \nvdash \sim\mathscr{P}$.

> Suppose T is a recursively axiomatized theory extending Q. Once again consider $\mathsf{stop}(\mathsf{i},\mathsf{n},\mathsf{j})$; since stop is recursive and T extends Q, stop is captured in T by some $Stop(i,n,j,z)$; let $\mathscr{H}(\bar{\mathsf{i}}) = \exists z Stop(\bar{\mathsf{i}}, \bar{\mathsf{i}}, z, \emptyset)$, and consider a Turing machine $\Pi_{\mathsf{s}}(\mathsf{i})$ which for arbitrary i, tests whether successive values of m number a proof of $\sim\mathscr{H}(\bar{\mathsf{i}})$, halting if some m numbers a proof and otherwise continuing forever — so $\Pi_{\mathsf{s}}(\mathsf{i})$ evaluates $\textsc{prft}(\mathsf{m}, \ulcorner\sim\mathscr{H}(\bar{\mathsf{i}})\urcorner)$, for successive values of m;[4] since T is a recursively axiomatized theory, this is a recursive relation so that there must be some such Turing machine. We can think of $\Pi_{\mathsf{s}}(\mathsf{i})$ as seeking a proof that $\Pi_{\mathsf{i}}(\mathsf{i})$ does not halt.
>
> Suppose $\Pi_{\mathsf{s}}(\mathsf{s})$ halts. By definition, $\Pi_{\mathsf{s}}(\mathsf{i})$ halts just in case some m numbers a proof of $\sim\mathscr{H}(\bar{\mathsf{i}})$; since $\Pi_{\mathsf{s}}(\mathsf{s})$ halts, then, there is some m such that $\textsc{prft}(\mathsf{m}, \ulcorner\sim\mathscr{H}(\bar{\mathsf{s}})\urcorner)$; so $T \vdash \sim\mathscr{H}(\bar{\mathsf{s}})$. But if $\Pi_{\mathsf{s}}(\mathsf{s})$ halts, for some m, $\langle\langle \mathsf{s},\mathsf{s},\mathsf{m}\rangle, \mathsf{0}\rangle \in \mathsf{stop}$; so by capture, $T \vdash Stop(\bar{\mathsf{s}}, \bar{\mathsf{s}}, \bar{\mathsf{m}}, \emptyset)$; so $T \vdash \exists z Stop(\bar{\mathsf{s}}, \bar{\mathsf{s}}, z, \emptyset)$, which is to say, $T \vdash \mathscr{H}(\bar{\mathsf{s}})$. So if T is consistent, $\Pi_{\mathsf{s}}(\mathsf{s})$ does not halt.
>
> (i) Suppose T is consistent and $T \vdash \sim\mathscr{H}(\bar{\mathsf{s}})$; then for some m, $\textsc{prft}(\mathsf{m}, \ulcorner\sim\mathscr{H}(\bar{\mathsf{s}})\urcorner)$; so by its definition, $\Pi_{\mathsf{s}}(\mathsf{s})$ halts ; but if T is consistent, $\Pi_{\mathsf{s}}(\mathsf{s})$ does not halt; so $T \nvdash \sim\mathscr{H}(\bar{\mathsf{s}})$.
>
> (ii) Suppose T is ω-consistent and $T \vdash \sim\sim\mathscr{H}(\bar{\mathsf{s}})$; then $T \vdash \mathscr{H}(\bar{\mathsf{s}})$; so $T \vdash \exists z Stop(\bar{\mathsf{s}}, \bar{\mathsf{s}}, z, \emptyset)$. But since $\Pi_{\mathsf{s}}(\mathsf{s})$ does not halt, for any m, $\langle\langle \mathsf{s},\mathsf{s},\mathsf{m}\rangle, \mathsf{0}\rangle \notin \mathsf{stop}$; and by capture, for any m, $T \vdash \sim Stop(\bar{\mathsf{s}}, \bar{\mathsf{s}}, \bar{\mathsf{m}}, \emptyset)$; so by ω-consistency, $T \nvdash \exists z Stop(\bar{\mathsf{s}}, \bar{\mathsf{s}}, z, \emptyset)$. This is impossible, $T \nvdash \sim\sim\mathscr{H}(\bar{\mathsf{s}})$

Again, this is roughly the form in which Gödel first proved the incompleteness of arithmetic. However, as we have seen it is possible to strengthen this version of the result to drop the requirement of ω-consistency for the simple result that no consistent, recursively axiomatizable theory extending Q is negation complete.

E14.9. Use the definition problem for μ-recursive functions to show that there is no μ-recursive enumeration of the set of truths of $\mathscr{L}_{\text{NT}}$. Hint: Return to $\mathsf{val}(\mathsf{i},\mathsf{n})$,

[4] Or, rather, since it has i free but numbers a formula with $\bar{\mathsf{i}}$ for x, the second term is $\textsc{formsub}(\ulcorner\sim\mathscr{H}(x)\urcorner, \ulcorner x\urcorner, \mathsf{num}(\mathsf{i}))$. See p. 682 and p. 728 below.

$Val(i, n, y)$ and $Def(\bar{\mathsf{i}})$ (this time depending on T12.7 for the result that Val expresses val). Suppose there is an enumeration $\mathsf{ent(n)}$ of the truths of $\mathscr{L}_{NT}$; then to get something that returns $\mathsf{0}$ and $\mathsf{1}$ in the right way, the characteristic function of $\mathsf{enf}\{\mu\mathsf{y}[\mathsf{ent(y)} = \ulcorner Def(\bar{\mathsf{i}})\urcorner \lor \mathsf{ent(y)} = \ulcorner {\sim}Def(\bar{\mathsf{i}})\urcorner]\} = \ulcorner Def(\bar{\mathsf{i}})\urcorner$ is $\mathsf{0}$ when the minimization finds $Def(\bar{\mathsf{i}})$ in the enumeration, and otherwise $\mathsf{1}$.

E14.10. Use your results for μ-recursive functions from other exercises to show that if T is a recursively axiomatized theory extending $\mathrm{Q_s}$, then there is a sentence $\mathscr{P}$ such that if T is consistent $T \nvdash \mathscr{P}$, and if T is ω-consistent, $T \nvdash {\sim}\mathscr{P}$.

14.3 Church's Thesis

We have attained a number of negative results, as T14.6 that if Q is ω-consistent then no Turing computable function $\mathsf{f(n)}$ returns zero just for numbers of theorems of predicate logic, and T14.7 that the set of truths of $\mathscr{L}_{NT}$ is not recursively enumerable. These are interesting. But, one might very well think, if no Turing machine computes a function, then we ought simply to compute the function some *other* way. So the significance of our negative results is magnified if the Turing computable functions are, in some sense, the *only* computable functions. If in some important sense the Turing computable functions are the only computable functions, and no Turing machine computes a function, then in the relevant sense the function is not computable. Thus Church's Thesis:

CT The total numerical functions that are effectively computable by some algorithmic method are just the recursive functions.

We want to be clear first, on the *content* of this thesis, and once we know what it says on reasons for thinking that it is true.

14.3.1 The content of Church's thesis

Church's thesis makes a claim about "total numerical functions that are effectively computable by an algorithmic method." Original motivations for computation of this sort are from the simple routines we learn in grade school for addition, multiplication, and the like. By themselves, such methods are of interest. However, we mean to include the sorts of methods contemporary computing devices can execute. These

are of considerable interest as well. Let us take up the different elements of the proposal in turn.

First, as always, a numerical function is *total* iff it is defined on the entire numerical domain. Arbitrary functions on a finite domain may be finitely specified by listing their members, and then computed by simple lookup. This was our approach with simple, but arbitrary, functions from chapter 4. The question of comuputability becomes interesting when domains are not finite (and from methods like those in the countability reference a function on an infinite domain is always comparable to one that is total). So Church's thesis is a thesis about the computability of total functions.

A function is *effectively computable* iff there is a method for finding its value for any given argument. Correspondingly, a property or relation is *effectively decidable* iff its characteristic function is effectively computable. So methods for addition and multiplication are adequate to calculate the value of the function for any inputs. Or consider a Turing machine programmed to enumerate the theorems of T, stopping with output 0 if it reaches (the number for) $\mathscr{P}$, and output 1 if it reaches $\sim\mathscr{P}$. If T is a consistent recursively axiomatized and negation complete theory, then this is an effective method for deciding the theorems of T. If $\mathscr{P}$ is a theorem, it eventually shows up in the enumeration, and the Turing machine stops with output 0. If $\mathscr{P}$ is not a theorem, $\sim\mathscr{P}$ is a theorem, so $\sim\mathscr{P}$ eventually shows up in the enumeration, and the machine stops with output 1. This was the idea behind T12.18. But if T is not negation complete, this is not an effective method for deciding theorems of T. If $\mathscr{P}$ is a theorem, it eventually shows up in the enumeration, and the machine stops with output 0. But if T is not negation complete, $\sim\mathscr{P}$ might fail to be a theorem. In this case, the machine continues forever, and does not stop with output 1; so for some arguments, this method does not find the value of the characteristic function, and we have not described an *effective* method for deciding the theorems of this T.

From the start, we may agree that there is some uncertainty about the notion of an *algorithmic* method; so, for example, different texts offer somewhat different definitions. However, as we did for logical validity and soundness in chapter 1, we shall take a particular account as a technical definition — partly as clarified in examples that follow. Difficulties to the side, there does seem to be a relevant core notion: for our purposes an *algorithmic* method is a finitely constrained rule-based procedure (rote, if you will).[5]

There is some vagueness in how much "processing" is allowed in following a rule. (So, an algorithm for multiplication does not typically include instructions for required additions.) However, we may take it that if some instructions are suf-

[5] We have no intention of engaging Wittgenstenian concerns about following a rule. See, for example, Kripke *Wittgenstein on Rules and Private Language*.

ficient for a computer to calculate a function, then the function is algorithmically computable. Thus that a function is Turing computable is sufficient to show that it is algorithmically computable. Again, standard methods for addition and multiplication are examples of algorithmic procedures. Truth table construction is another example of a method that proceeds by rote in this way. Given the basic tables for the operators, one simply follows the rules to complete the tables and determine validity — and one could program a computer to perform the same task. Thus validity in sentential logic is effectively decidable by an algorithmic method. In contrast, derivations are not an algorithmic method. The strategies are helpful! But, at least in complex cases, there may come a stage where insight or something like lucky guessing is required. And at such a stage, you are not following any rules by rote, and so not following any specific algorithm to reach your result.

And algorithmic methods operate under finite constraints. In general, we shall not worry about how large these constraints may be, so long as they remain finite. Consider first, truth table construction. If this is to be an effective method for determining validity, it should return a result for any sentence. But for any $\mathsf{n} > 0$ there are sentences with that many atomic sentences (for example, $A_1 \wedge A_2 \wedge \ldots \wedge A_n$), so the corresponding table requires 2^{n} rows. This number may be arbitrarily large — and a table may require more paper or memory than in the entire universe. But, in every case, the limit is finite. So, for our purposes, it qualifies as an effective algorithmic method. Contrast this case with a device, which we may call "god's mind," that stores all the theorems of predicate logic sorted in order of their Gödel numbers. To calculate whether $\mathcal{P}$ is a theorem, simply search up to the Gödel number of $\mathcal{P}$ to see if that sentence is in the database: if it is, $\mathcal{P}$ is a theorem, if it is not $\mathcal{P}$ is not a theorem. Alternatively, perhaps this machine does infinite parallel processing, and for every n runs a Turing machine to evaluate $\textsc{prfpl}(\mathsf{n}, \ulcorner\mathcal{P}\urcorner)$ "all at the same time" as it were — so that if some calculation evaluates to $\mathsf{0}$, $\mathcal{P}$ is a theorem, and if all evaluate to $\mathsf{1}$, $\mathcal{P}$ is not. It is not our intent to deny the existence of god, or that one might very well solve mathematical problems by prayer (though this might not go over very well on examinations which require that you show your work)! But, insofar as this device requires infinite memory, infinitely many instructions, infinite processing power, or the capacity to evaluate at once infinite ranges of data, it will not, for our purposes count as an algorithmic method.

Or consider again a Turing machine programmed to enumerate the theorems of T, stopping with output $\mathsf{0}$ if it reaches (the number for) $\mathcal{P}$, but continuing forever if $\mathcal{P}$ does not appear. One might suppose the information that $\mathcal{P}$ is not a theorem is contained already in the fact *that the machine never halts*, and that god or some being with an infinite perspective might very well extract this information from the

machine. Perhaps so. But this method is not algorithmic just because it requires the infinite perspective. But there are interesting attempts to attain the effect of this latter machine without appeals to god. Consider, first, "Zeno's machine." As before, the machine enumerates theorems, this time flashing a light if $\mathscr{P}$ appears in the list. However, for some finite time t (say 60 seconds), this machine takes its first step in $t/2$ seconds, its second step in $t/4$ seconds, and for any n, step n in $t/2n$ seconds. But the sum of $t/2 + t/4 + \ldots = t$, and the Turing machine runs through all of infinitely many steps in time t. So start the machine. If the light flashes before t seconds elapse, $\mathscr{P}$ is a theorem. If t elapses, the machine has run through all of infinitely many steps, so if the light does not flash, $\mathscr{P}$ is not a theorem.

One might object this proposal reduces to a tautology of the sort, "If such-and-such (impossible) circumstances obtain, then the theorems are decidable." Great, but who cares? However, we should not reject the general strategy out-of-hand. From even a very basic introduction to special relativity, one is exposed to time dilation effects (for a simple case see the time dilation reference). General relativity allows a related effect. Where special relativity applies just to reference frames moving at constant velocity relative to one another, general relativity allows accelerated frames. And it is at least consistent with the laws of general relativity for one frame to have an infinite elapsed time, while another's time is finite.[6] So, for a Malament-Hogarth (M-H) machine, put a Turing machine in the one frame and an observer in the other. The Turing machine operates in the usual way in its frame enumerating the theorems forever. If $\mathscr{P}$ is a theorem, it sends a signal back to the observer's frame that is received within the finite interval. From the observer's perspective, this machine runs through infinitely many operations. So if a signal is received in the finite interval, $\mathscr{P}$ is a theorem. If no signal is received in the finite interval, then $\mathscr{P}$ is not a theorem. (And similarly, the M-H machine might search for a counterexample to the Goldbach conjecture, or the like.) There is considerable room for debate about whether such a machine is physically possible. But, even if physically realized, it is not *algorithmic*. For we require that an algorithmic method terminates in a finite number of steps.

Church's thesis is thus that the total numerical functions that are effectively computable by some algorithmic method are the the same as the recursive functions. Suppose we obtain a negative result that some function is not algorithmically computable. Even with the finite limits we have placed on memory, number of instructions and the like, the negative result remains of considerable interest: So long as a routine follows

[6]Students with the requisite math and physics background might be interested in Hogarth, "Does General Relativity Allow an Observer To View an Eternity In a Finite Time?" See also Earman and Norton, "Forever is a Day," and for the same content, chapter 4 of Earman, *Bangs, Crunches, Whimpers, and Shrieks* (but with additional, though still difficult, setup in earlier chapters of the text).

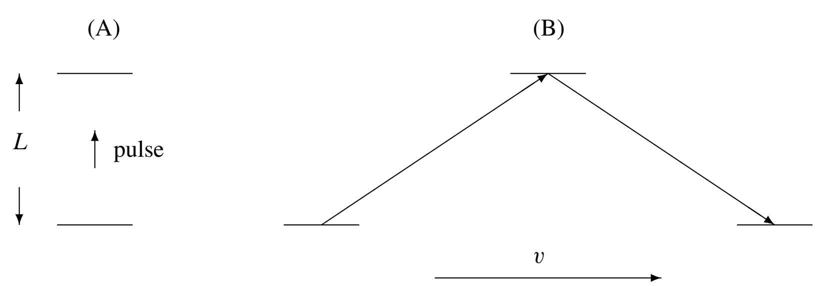
Simple Time Dilation

It is natural to think that, just as a wave in water approaches a boat faster when the boat is moving is moving toward it than when the boat is moving away, so light would approach an observer faster when she is moving toward it, and more slowly whens she is moving away. But this is not so. The 1887 Michelson-Morley experiment (and many others) verify that the speed of light has the *same* value for all observers. Special relativity takes as foundational:

1. The laws of physics may be expressed in equations having the same form in all frames of reference moving at constant velocity with respect to one another.

2. The speed of light in free space has the same value for all observers, regardless of their state of motion.

These principles have many counterintuitive consequences. Here is one: Consider a clock which consists of a pulse of light bouncing between two mirrors separated by distance L as in (A) below. Where c is the constant speed of light, the time between ticks is the distance traveled by the pulse divided by its speed L/c.



Now consider the same clock as observed from a reference frame relative to which it is in motion, as in (B). The speed of light remains c (instead of being increased, as one might expect, by the addition of the horizontal component to its velocity). But the distance traveled between ticks is greater than L, so the time between ticks is greater than L/c — which is to say the clock ticks more slowly from the perspective of the second frame.

One might wonder happens if this clock is rotated 90 degrees so that the pulse is bouncing parallel to the direction of motion, or what would happen if time were measured by a pendulum clock. But within a frame, everything is coordinated according to the usual laws: On special relativity, there are coordinated changes to length, mass and the like so that the effect is robust. As observed from a reference frame relative to which the frame is in motion, time, mass, and length are distorted together. For further discussion, consult any textbook on introductory modern physics.

definite rules, no (finite) amount of parallel processing, high-speed memory and so forth is going to make a difference — the function remains uncomputable.

14.3.2 The basis for Church's thesis

It is widely accepted that Church's thesis is true, but also that it is not susceptible to *proof*. We shall return to the question of proof. There are perhaps three sorts of reasons that have led philosophers, computer scientists and logicians to think it is true. (i) A number of independently defined notions plausibly associated with computability converge on the recursive functions. (ii) No plausible counterexamples — algorithmically computable functions not recursive, have come to light. And (iii) there is a sort of rationale from the nature of an algorithm. This last may verge on, or amount to, demonstration of Church's thesis.

Independent definitions. We have already seen that the Turing computable functions are the same as the recursive functions. And we are in a position to close another loop. From T12.16, any recursive function is captured by a recursively axiomatized consistent theory extending Q. But also,

T14.10. Every (total) function that can be captured by a recursively axiomatized consistent theory extending Q is recursive.

Suppose a function $\mathsf{f(m) = n}$ can be captured in a recursively axiomatized consistent theory T extending Q; then there is some $\mathscr{F}(x, y)$ such that if $\langle\mathsf{m,n}\rangle \in \mathsf{f}$, then $T \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ and if $\langle\mathsf{m,n}\rangle \notin \mathsf{f}$ then $T \vdash \sim\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$; and from the latter, since T is consistent, $T \nvdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. But since f is a function, if $\langle\mathsf{m,n}\rangle \in \mathsf{f}$, any $\mathsf{k \neq n}$ is such that $\langle\mathsf{m,k}\rangle \notin \mathsf{f}$; so that $T \nvdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{k}})$. So if $\langle\mathsf{m,n}\rangle \in \mathsf{f}$ then (i) for $\mathsf{b} = \ulcorner\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})\urcorner$ there is some a such that $\textsc{prft}(\mathsf{a,b})$; and (ii) n is the only (and so least) number such that $T \vdash \mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$.

Intuitively, we can find the value of f(m) by searching the theorems until we find one of the sort $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$; and from this derive the value n. More formally: First, for the number of $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$,

$$\mathsf{numf(m,n)} =_{\text{def}} \mathsf{formsub[formsub(}\ulcorner\mathscr{F}(x,y)\urcorner, \ulcorner x\urcorner, \mathsf{num(m)}), \ulcorner y\urcorner, \mathsf{num(n)}]$$

Recall that formsub(p, v, s) takes the Gödel numbers of a formula $\mathscr{P}$, variable x and term s and returns the number of $\mathscr{P}^{x}_{s}$; and num(m) returns the Gödel number of the standard numeral for m. So this gives the Gödel number of $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ as a function of m and n. Now since T is recursively axiomatized and extends Q there is a recursive $\textsc{prft}$ and,

$$\mathsf{f(m)} =_{\text{def}} \mathsf{exp}(\mu \mathsf{z}[\mathsf{len(z)} = 2 \wedge \textsc{prft}(\mathsf{exp(z,0)}, \mathsf{numf(m, exp(z,1))})], 1)$$

So z is of the sort $2^{\mathsf{a}} \times 3^{\mathsf{n}}$, where a numbers a proof of $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$; that is, $\mathsf{exp(z,0)}$ numbers a proof of $\mathsf{numf(m, exp(z,1))}$. But there is only one n that could result in a proof of $\mathscr{F}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$. And n is easily recovered from z. So $\mathsf{f(m)}$ is a recursive function.

So a function is captured in a recursively axiomatized consistent theory extending Q iff it is recursive. So these three independently defined notions for computing functions are extensionally equivalent.[7] And increasing the power of a deductive system from Q to PA and beyond does not extend the range of captured functions.

E14.11. (i) Explain how the result that the constructed $\mathsf{f(m)}$ in T14.10 is recursive requires that the original function captured by $\mathscr{F}(x, y)$ is *total*. (ii) Explain how the result changes in case we drop the requirement that captured functions be total. Hint: The construction still works, but the result is μ-recursive, not recursive.

Failure of counterexamples. Another reason for accepting Church's thesis is the failure to find counterexamples. This may be very much the same point as before: When we set out to define a notion of computability, or compute a function, what we end up with are recursive functions, rather than something other. Of course, god's mind, Zeno's machine, an M-H machine, or the like might compute a non-recursive function. Perhaps there are such devices. However, on our account, they are not algorithmic. What we do not seem to have are algorithmic methods for computing non-recursive functions.

But also in this category of reasons to accept Church's thesis is the failure of a natural strategy for showing that Church's thesis is false. Suppose one were to to propose that the *primitive* recursive functions are all the computable functions, and so that regular minimization is redundant (perhaps you have had this very idea). Here is a way to see this hypothesis false: Observe that the primitive recursive functions are recursively enumerable. For this, treat composition and recursion as operations on functions so that,

$$\mathsf{plus(x,y)} =_{\text{def}} \mathsf{Rec}[\mathsf{zero(x)}, \mathsf{Comp}(\mathsf{suc(x)}, \mathsf{idnt}^3_3(\mathsf{x,y,u}))]$$

[7] And there are more. Church himself was originally impressed by an equivalence between his *lambda calculus* and the recursive functions. As additional examples, Markov algorithms are discussed in Mendelson, *Introduction to Mathematical Logic*, §5.5; abacus machines in Boolos et al., *Computability and Logic*, §5; see below for discussion of the Kolmogorov-Uspenskii machine.

And so forth. Then assign numbers in the usual way,

a.	$g[,] = 3$	f.	$g[\mathsf{Comp}] = 13$
b.	$g[)] = 5$	g.	$g[\mathsf{Rec}] = 15$
c.	$g[(] = 7$	h.	$g[\mathsf{Min}] = 17$
d.	$g[\mathsf{zero}] = 9$	i.	$g[\mathsf{x_i}] = 19 + 4i$
e.	$g[\mathsf{suc}] = 11$	j.	$g[\mathsf{idnt}^j_k] = 21 + 4(2^j \times 3^k)$

Min does not matter for primitive recursive functions, but is included for later. The most important element of the construction is a PRSEQ(m, n) which, on the model of FORMSEQ from chapter 12, identifies precursors to a primitive recursive function as initial functions or formed from ones before by composition or recursion. From that, there is a relation PR(n) true of numbers for primitive recursive functions. And there is an enumeration, $\mathsf{epr}(0) =_{\text{def}} \mu z[\text{PR}(z)]$, and $\mathsf{epr}(Sy) =_{\text{def}} \mu z[z > \mathsf{epr}(y) \land \text{PR}(z)]$. So there is a recursive enumeration of the primitive recursive functions, there is an enumeration of the functions of one free variable, and so forth. Details are left for an exercise.

So consider an enumeration of the primitive recursive functions of one free variable and an array as follows.

(K)

	0	1	2	...
f_0	$\mathbf{f_0(0)}$	$f_0(1)$	$f_0(2)$	
f_1	$f_1(0)$	$\mathbf{f_1(1)}$	$f_1(2)$	
f_2	$f_2(0)$	$f_2(1)$	$\mathbf{f_2(2)}$	
$\vdots$				

And consider the function $d(n) = f_n(n) + 1$. This function is *computable*; for any n: (i) run the enumeration to find f_n; (ii) run f_n to find $f_n(n)$; (iii) add one. Since each step is recursive, the whole is computable. But d(n) is not primitive recursive: $d(0) \neq f_0(0)$; $d(1) \neq f_1(1)$; and in general, $d(n) \neq f_n(n)$; so d is not identical to any of the primitive recursive functions. So there are computable functions that are not primitive recursive.

It is natural to think that a related argument would show that not all computable functions are recursive: recursively enumerate the recursive functions; then diagonalize to find a computable function not on the list. But this does not work! It is is an entirely "grammatical" matter to identify the primitive recursive functions — the function epr(n) results purely as a matter of form. But there is no parallel method for the recursive fuctions. In homework (E12.7, E12.16, E14.7) we have introduced the *μ-recursive* functions. These are like the recursive functions but without the regularity requirement for minimization. So all the recursive functions are μ-recursive, but some μ-recursive functions are not recursive. Where every recursive function $f(\vec{x})$ is

total in the sense that it returns a value for every $\vec{x}$, some μ-recursive functions are *partial* insofar as there may be values of $\vec{x}$ for which they return no value (as occurs when minimization is applied to a $g(\vec{x}, y)$ that never evaluates to zero). By a simple extension of the reasoning from above, there is an enumeration of μ-recursive functions f_i. Again enumeration reverts to a purely grammatical matter. But from E14.7 there is a definition problem, like the halting problem, according to which there is no μ-recursive function def_i such that $def(i) = 0$ if $f_i(i)$ is defined and $def(i) = 1$ if $f_i(i)$ is undefined. And from this there cannot be a recursive means of saying when a minimization operation "halts," and so when a function is *regular* — for a program to pick out regular functions would solve a version of this definition problem.

> For any μ-recursive function $f(x)$, $\mu y[y = f(x)]$ is a μ-recursive function equivalent to it. So we simply suppose that μ-recursive functions can always be cast in this form, and consider an enumeration of the μ-recursive functions of a single free variable. Consider $f_i(x) = \mu y[g(x, y)]$ and $f_j(y) = g(i, y)$. Suppose some μ-recursive function $reg(i)$ returns zero when i numbers a regular function and is otherwise 1. A function of one variable is regular just in case its minimization is defined; so $reg(j)$ iff $\mu y[g(i, y)] = f_i(i)$ is defined; iff $def(i)$. So for any i there is a j such that $reg(j)$ iff $def(i)$; so reg is sufficient to solve the definition problem; reject the assumption.

So we are blocked from recursively enumerating the recursive functions, and so from this means of finding a computable function that is not a recursive function.

*E14.12. (i) Clean up and complete the reasoning to show that there is a recursive enumeration of the primitive recursive functions; [extended hints – see dump in answers] that is, find $rvar(n)$, $rvec(n)$, $new(j, n)$ and then $\textsc{pr}(n)$. (ii) For any (primitive) recursive function $f(x)$ there is a canonical formula $\mathscr{F}(x, y)$ to capture it in theories extending Q. Thus the enumeration $eprf(n)$ of primitive recursive functions extends to an enumeration $eprc(n)$ whose value is the number of the formula to capture $eprf(n)$. Given this enumeration, extend the construction from T14.10 to find the (recursive) function that is (Turing) computable but not primitive recursive. Hint: You will be able to construct a function $valpr(i, m)$ to return the the value of $f_i(m)$ and use this for the final result.

E14.13. (i) Extend the demonstration that the primitive recursive functions are enumerable to show that there is an enumeration of the μ-recursive functions. (ii)

From E12.16 for any μ-recursive function $\mathsf{f(x)}$ there is a canonical formula $\mathscr{F}(x, y)$ to capture it in theories extending $\mathrm{Q_s}$; thus, again, your enumeration $\mathsf{emrf(n)}$ of μ-recursive functions extends to an enumeration $\mathsf{emrc(n)}$ whose value is the number of the formula to capture $\mathsf{emrf(n)}$. Given this enumeration, extend the construction from T14.10 to find a μ-recursive $\mathsf{val(i, n)} = \mathsf{f_i(n)}$.

The nature of an algorithm. There are also reasons for Church's thesis from the very nature of an algorithm.[8] Perhaps the "received wisdom" with respect to Church's thesis is as follows.

> The reason why Church's [Thesis] is called a *thesis* is that it has not been rigorously proved and, in this sense, it is something like a "working hypothesis." Its plausibility can be attested inductively — this time not in the sense of mathematical induction, but "on the basis of particular confirming cases." The Thesis is corroborated by the number of intuitively computable functions commonly used by mathematicians, which can be defined within recursion theory. But Church's Thesis is believed by many to be destined to *remain* a thesis. The reason lies, again, in the fact that the notion of effectively computable function is a merely intuitive and somewhat fuzzy one. It is quite difficult to produce a completely rigorous proof of the equivalence between intuitively computable and recursive functions, precisely because one of the sides of the equivalence is not well-defined (Berto, *There's Something About Gödel*, pp. 76-77.)

There are a couple of themes in this passage. First, that Church's thesis is typically accepted on grounds of the sort we have already considered. Fair enough. But second that it is not, and perhaps cannot, be proved. The idea seems to be that the recursive functions are a precise mathematically defined class, while the algorithmically computable functions are not. Thus there is no hope of a demonstrable equivalence between the two.

But we should be careful. Granted: If we start with an inchoate notion of computable function that includes, at once, calculations with pencil and paper, calculations on the latest and greatest supercomputer, and calculations on Zeno's machine, there will be no saying whether the computable functions definitely are, or are not, identical to the Turing computable functions. But this is not the notion with which

[8]Material in this section is developed from Smith, *An Introduction to Gödel's Theorems*, chapter 45; Smith, "Squeezing Arguments"; along with Kolmogorov and Uspenskii, "On the Definition of an Algorithm." See also Black, "Proving Church's Thesis."

we are working. We have a relatively refined technical account of algorithmic computability. Of course, it is not yet a *mathematical* definition. But neither are our chapter 1 accounts of logical validity and soundness; yet we have been able to show in T9.1 that any argument that is quantificationally valid (in our mathematical sense) is logically valid. And similarly, the whole translation project of chapter 5 assumes the possibility of moving between ordinary and mathematical notions. It is at least possible that a vaguely defined predicate might pick out a precise object ("the number of people on campus," on a university with a core campus area and other empty but vaguely associated land, might be 15,214 despite vagueness in "campus"). The question is whether we can "translate" the notion of an algorithm to formal terms.

So let us turn to the hard work of considering whether there is an argument for accepting Church's thesis. A natural first suggestion is that the step-by-step and finite nature of any algorithm is always within the reach of, or reflected by, some Turing program or recursive function, so that the algorithmically computable functions are inevitably recursively computable.[9] Already, this may amount to a consideration or reason in favor of accepting the Thesis. In chapter 45 of his *An Introduction to Gödel's Theorems*, Peter Smith advances a proposal according to which such considerations amount to proof.

Smith's overall strategy involves "squeezing" algorithmic computability between a pair of mathematically precise notions. Even if a condition C (say, "being a tall person") is vague, it might remain that there is some completely precise sufficient condition S (being over seven feet tall), such that anything that is S is C, and perfectly precise necessary condition N (being over five feet tall) such that anything that is C is N. So,

$$S \quad \Longrightarrow \quad C \quad \Longrightarrow \quad N$$

If it should also happen that N implies S, then the loop is closed, so that,

$$S \quad \Longleftrightarrow \quad C \quad \Longleftrightarrow \quad N$$

And the target condition C is equivalent to (squeezed between) the precise necessary and sufficient conditions. Of course, in our simple example, N does not imply C: being over five feet tall does not imply being over seven feet tall.

For Church's thesis, we already have that Turing computability is sufficient for algorithmic computability. So what is required is some necessary condition so that,

$$T \quad \Longrightarrow \quad A \quad \Longrightarrow \quad N$$

[9]This idea is contained already in the foundational papers of Church, "An Unsolvable Problem," and Turing, "On Computable Numbers."

Turing computability implies algorithmic computability and algorithmic computability implies the necessary condition. Church's thesis follows if, in addition, N implies Turing computability. As it turns out, we shall be able to specify a condition N which (mathematically) implies T. For demonstration of Church's thesis, it will be more controversial whether A implies N.

The argument has three stages: The idea is that, (i) there are some necessary features of an algorithm, such that any algorithm has those features; (ii) any routine with those features is embodied in a generalized version of a Turing machine, a Kolmogorov-Uspenskii (K-U) machine; (iii) every function that is K-U computable is recursive, and so Turing computable.

$$\boxed{\textit{Necessary features}} \longrightarrow \boxed{\textit{K-U computability}} \longrightarrow \boxed{\textit{Turing computability}}$$

The result is that K-U computability works as as the precise condition N in the squeezing argument: A implies N, and N implies T. So T iff A iff N, and Church's thesis is established — or no less plausible than is the conclusion of this argument.

Perhaps the following are necessary conditions on any algorithm. We are are free, however, to hold that any routine which satisfies the constraints is an algorithm; if this is so the conditions are necessary and sufficient, and we may see them as an extension of our initial more sketchy account.[10]

AC (1) There is some *dataspace* consisting of a finite array of "cells" which may stand in some relations R_0, $R_1 \ldots R_a$ and contain some symbols $s_0, s_1 \ldots s_b$.

(2) At every stage in a computation, there is some finite "active" portion of the dataspace upon which the algorithm operates.

(3) The body of the algorithm includes finitely many instructions for modifying the active portion of the dataspace depending on its character, and for jumping to the next set of instructions.

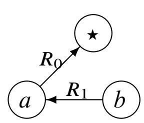
(4) For the calculation of a function $\mathsf{f}(\vec{\mathsf{x}}) = \mathsf{y}$ there is some finite initial representation of $\vec{\mathsf{x}}$ and some way to read off the value of y, after a finite number of steps.

[10] Smith seems to grant that some such conditions are necessary, even though some method may satisfy the conditions yet fail to count as an algorithm. Perhaps this is because he is impressed by the initial examples of routines implemented by human agents with relatively limited computing power. This is not a problem for his squeezing argument, since the corresponding recursive function may yet be computable by some other method which satisfies more narrow constraints — for example, by a Turing machine.

So this sets up an algorithm abstractly described. It is hard to see how an algorithm would not involve some space, portions of which would stand in different relations. At any given stage, the algorithm operates on some portion of the space, where these operations may depend upon, and modify the arrangement of the active space. The algorithm itself consists of some instructions for operating on the dataspace, where these are generically of the sort, "if the active area is of type t, perform action a, and go to new instructions q." The calculation of a function $\mathsf{f}(\vec{\mathsf{x}})$ somehow takes $\vec{\mathsf{x}}$ as an input, and gives a way to read off the value of y as an output. And an algorithm terminates in a finite number of steps. The finite constraints on the dataspace, relations, symbols and area from (1) and (2) seem to be consequences of (3) and (4): There is some upper bound to the space modified by instructions from a finite collection, each member of which modifies at most a finite area. Then beginning with a finite initial representation of some $\vec{\mathsf{x}}$, including finitely many cells of the dataspace standing in finitely many relations, filled with finitely many symbols and then modifying finite portions of the space finitely many times, all we are going to get are finitely many cells, standing in finitely many relations, filled with finitely many symbols.

On the face of it, given their extreme simplicity, it is not obvious that Turing machines compute every algorithmically computable function. But a related device, the K-U machine, described in 1958 (the cited English translation is from later) purports to implement conditions along these lines. A somewhat modified version of the original K-U machine is as follows.

KU (1) There are some cells c_0, $c_1 \dots c_a$ which may stand in binary relations R_0, $R_1 \dots R_b$ and contain symbols s_0, $s_1 \dots s_c$. In simple cases, we may think of such arrangements graphically as follows, where different relations are represented by arrows of different colors.

(L) 

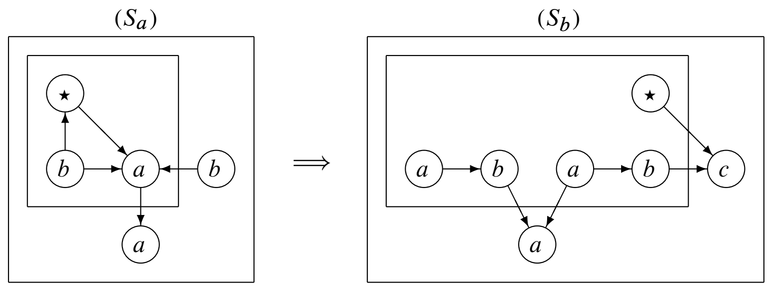
Each arrow, regardless of direction is an *edge*.

(2) The dataspace always includes exactly one "origin" whose content is some arbitrary symbol as $\star$ in the the upper cell of (L) — where the active area includes all cells on paths $\leq n$ edges from the origin, for $n \geq 1$.

(3) Instructions are finitely many quadruples of the sort $\langle q_i, S_a, S_b, q_j \rangle$ where q_i and q_j are instruction labels; S_a describes an active area; and S_b a state with which the active area is to be replaced. Associate each cell in S_a with the least number of edges between it and the origin; let n be the

greatest such integer in S_a; this n remains the same in every quadruple with label q_i, though the value of n may vary as q_i varies. Again, instructions are a function in the sense that no instruction has $\langle q_i, S_a\rangle$ the same but $\langle S_b, q_j\rangle$ different. We may see S_a and S_b as follows.

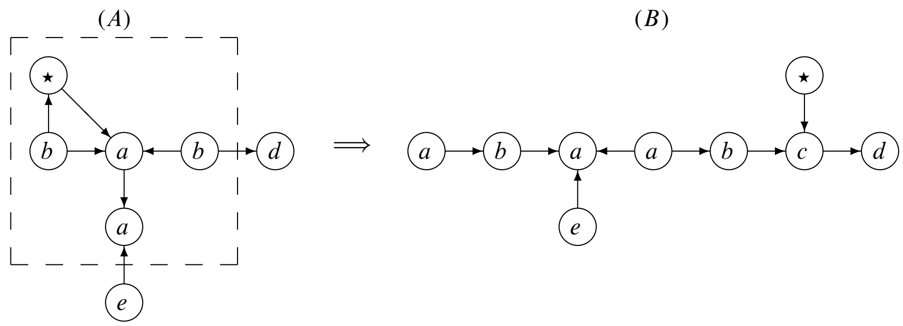
(M)



In this case $n = 2$. The active area S_a is replaced by the configuration S_b. The concentric rectangles indicate the "boundary" cells which may themselves be related to cells not part of the active area; the replacing area must have a boundary with cells to match boundary cells of the active area.

(4) There is some finite initial setup, and some means of reading off the final value of the function (for different relation and symbol sets, these may be different). We think of the origin cell as the "machine head," where an algorithm always begins with an instruction label $q_i = 1$ and terminates when $q_i = 0$.

So a K-U machine is a significant generalization of a Turing machine. We allow arbitrarily many symbols. And the dataspace is no longer a tape with cells in a fixed linear relation, but a space with cells in arbitrary relations which may themselves be modified by the program. Instructions respond to, and modify, not just individual cells, but arbitrarily large areas of the dataspace. At the same time, a K-U machine remains a generalized *Turing machine:* It remains that an instruction q_i is of the sort, if S_a perform action A and go to instruction q_j. So, the instruction (M) might be applied to get,

(N) 

As indicated by the dotted line, the dataspace (A) has an active area of the sort required in instruction (M); so the active area is replaced according to the instruction to for the resultant space (B). The example is arbitrary. But that is the point: The machine allows arbitrary rote modifications of a dataspace. Observe that instructions with $S_a \neq S'_a$ might both map onto a given dataspace in case the number n of edges from the origin in S_a is different from S'_a (say an active area with a box for $n = 1$ inside the box in (N)). But the consistency requirement is satisfied with constant n: for consistency, it is sufficient to require that so long as $n(q_i, S_a)$ is a constant, there is no instruction with $\langle q_i, S_a\rangle$ the same but $\langle S_b, q_j\rangle$ different.

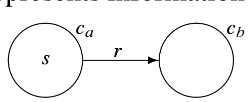
Perhaps the relation to Turing machines already makes it plausible that every K-U computable function is recursive. But we can argue for this result directly, very much as for T14.2.

T14.11. Every KU computable function is a recursive function.

We have been through this sort of thing a couple of times now, and I indicate only some of the key steps. (You will find further details in answers to E14.14 — though, of course, you should try it yourself!). Begin assigning numbers to labels, symbols, cells and relations in some reasonable way.

a. $g[q_i] = 3 + 8i$ c. $g[c_i] = 7 + 8i$
b. $g[s_i] = 5 + 8i$ d. $g[r_i^j] = 9 + 8(2^i \times 3^j)$

The number for an *edge*, $\mathsf{EDGE}(\mathsf{e})$ is of the sort $\pi_0^{\langle c_a\rangle} \times \pi_1^{\langle r_i^j\rangle} \times \pi_2^{\langle s_i\rangle} \times \pi_3^{\langle c_b\rangle}$, where the superscript for the relation is 1 or 0 depending on the direction of the arrow. Thus an edge represents information as follows,

(O) 

There are cells c_a and c_b related by r (in one direction or another) where c_a has content s. The edge leaves content of c_b undetermined, though it would be filled by

an edge in which that cell were the first. Then some *data*, $\mathrm{DATA}(d)$ is a sequence of edges $\pi_0^{e_a} \times \ldots \times \pi_n^{e_b}$. Cells m and n are *connected* on d just when edges beginning with the one reach to the other; that is, when there is a sequence with members from d, beginning with m, ending with n, where the ends are linked by intermediate members.

$$\begin{aligned}\mathrm{CONNECTED}(d,m,n) =_{\mathrm{def}}\ & (\exists x \leq d)\{(\forall i < len(x))(\exists j < len(d))[exp(x,i) = exp(d,j)] \wedge \\ & [exp(exp(x,0),0) = m \wedge exp(exp(x,len(x)-1),3) = n] \wedge \\ & (\forall i, 0 \leq i < len(x)-1)[exp(exp(x,i),3) = exp(exp(x,i+1),0)]\}\end{aligned}$$

So there is a sequence x with edges from d, whose first cell is numbered m and last cell numbered n, such that the last cell of one edge is the same as the first cell of the next. Then say a *dataspace*, $\mathrm{DATASP}(n)$ is some data every cell of which is connected to an origin cell 0, and no cell of which is connected back to itself (so connection in a dataspace is a strict partial order).

$$\begin{aligned}\mathrm{DATASP}(d) =_{\mathrm{def}}\ & \mathrm{DATA}(d) \wedge (\forall i < len(d))\mathrm{CONNECTED}(d, \langle 0\rangle, exp(exp(d,i),3)) \wedge \\ & \sim(\exists i < len(d))\mathrm{CONNECTED}(d, exp(exp(d,i),0), exp(exp(d,i),0))\end{aligned}$$

Then a *subspace* s of d, $\mathrm{SUBSP}(d,s)$ is a dataspace every link of which belongs to d. The *minimum* links to cell n, $minlnks(d,n)$ is the least y that is the length of a subspace connecting n to the origin. The *depth*, $depth(d)$ of a dataspace is the least y greater than or equal to the minimum number of links to every cell in the space. A *border* cell, $\mathrm{BORDER}(d,n)$ is a cell with $minlinks(d,n) = depth(d)$. The *n-space*, $nspace(d,n)$ is the least y including all the links in any subspace of d with depth n — so it includes all the cells in d up to depth n. And the *maximum* cell of a dataspace $maxcell(d)$ is the least y greater than or equal to every cell number in the space.

Where the cells are sequenced and numbered, spaces are most naturally comparable, not when they are identical, but when they are isomorphic. For this, a *pair*, $\mathrm{PAIR}(p)$ is of the sort $\pi_0^i \times \pi_1^j$; and a relation on a finite domain, $\mathrm{REL}(r)$ is a sequence $\pi_0^{p_0} \times \ldots \times \pi_n^{p_n}$. A relation is a 1:1 *map*, $\mathrm{MAP}(m)$ iff no x is related to more than one y, and different objects x are not related to the same y; so,

$$\begin{aligned}\mathrm{MAP}(m) =_{\mathrm{def}}\ & \mathrm{REL}(m) \wedge (\forall i < len(m))(\forall j < len(m))[\\ & exp(exp(m,i),0) = exp(exp(m,j),0) \leftrightarrow exp(exp(m,i),1) = exp(exp(m,j),1)]\end{aligned}$$

Map m has the cells of dataspace d in its domain, $\mathrm{DOM}(m,d)$ just in case m is a map (that takes 0 to 0 and) for any edge $\langle c_a, r_i^j, s_i, c_b\rangle$ in d, has some pair $\langle c_b, x\rangle$ in m. The output value of a map for a given input $mapv(m,x) = y$ for the least y such that $\langle x,y\rangle$ is in the map, and otherwise is some default value. Then dataspace b is a

projection of dataspace a on map m, $\mathsf{proj(m, a)} = \mathsf{b}$, just in case a and b are identical except that the cell numbers in a are mapped to cell numbers in b.

$$\begin{aligned}
&\mathsf{proj(m,a)} =_{\text{def}} \mu\mathsf{y}(\forall\mathsf{i} < \mathsf{len(a)})[\\
&\quad \mathsf{mapv(m, exp(exp(a,i),0))} = \mathsf{exp(exp(y,i),0)} \wedge \mathsf{exp(exp(a,i),1)} = \mathsf{exp(exp(y,i),1)} \wedge\\
&\quad (\mathsf{exp(exp(a,i),2)} = \mathsf{exp(exp(y,i),2)} \wedge \mathsf{mapv(m, exp(exp(a,i),3))} = \mathsf{exp(exp(y,i),3)}]
\end{aligned}$$

Spaces a and b *match* on map m, $\textsc{match}\mathsf{(m, a, b)}$ just in case each link in $\mathsf{proj(m, a)}$ is identical to a link in b and each link in b is identical to one $\mathsf{proj(m, a)}$. And spaces are *isomorphic* on a, $\textsc{iso}\mathsf{(a, b)}$ just in case there exists a map including domain a on which they so match (where the bound for the map is a function of the maximum cell numbers from the spaces.)

The number for an *instruction*, $\textsc{ins}\mathsf{(n)}$ is of the sort, $\pi_0^{\langle q_i\rangle} \times \pi_1^{\ulcorner S_a \urcorner} \times \pi_2^{\ulcorner S_b \urcorner} \times \pi_3^{\langle q_j\rangle}$, where any cell in the border of S_a reappears in S_b. And a K-U machine, $\textsc{kumach}\mathsf{(m)}$ is a sequence of instructions $\pi_0^{\mathsf{i}_0} \times \ldots \times \pi_\mathsf{n}^{\mathsf{i}_\mathsf{n}}$, where instructions at any label have the depth of S_a the same, but no instructions at the same label have S_a isomorphic. Then each K-U machine is associated with a Gödel number, and there is an enumeration of K-U machines. And from a K-U machine, instruction number, and dataspace, there is a function to machine states: the machine state is that instruction which for machine m has instruction label q_i, with S_a isomorphic to the same-sized portion of the dataspace d. As before, if a K-U machine includes states with instruction label q_i, but no instruction of the sort $\langle q_i, S_a, x, y\rangle$ let the machine be augmented to include $\langle q_i, S_a, S_a, q_i\rangle$; that way, it will "loop" rather than "hang" in that state. Then, $\mathsf{machs(m, q, d)} =$

$$\mu\mathsf{y}(\exists\mathsf{i} < \mathsf{len(m)})[\mathsf{y} = \mathsf{exp(m,i)} \wedge \mathsf{exp(y,0)} = \mathsf{q} \wedge \textsc{iso}\mathsf{(exp(y,1), nspace(d, depth(exp(y,1))))}]$$

So the machine state is the least y with label q such that S_a maps to the dataspace.

Now we are ready for recursive definitions $\mathsf{space(m, n, j)}$ and $\mathsf{state(m, n, j)}$ that describe the dataspace and machine state as a function of the K-U machine, initial value n of $\mathsf{f(n)}$, and step j of operation. Suppose functions $\mathsf{code(n)}$ and $\mathsf{decode(d)}$ to take the initial value n into a dataspace, and the final dataspace into the value it represents. We require an analog $\mathsf{d} \ominus \mathsf{a}$ to $\mathsf{a} \circ \mathsf{b}$ that takes a dataspace d, an active area a and returns d without a. For this, recursively define $\mathsf{del(d, a, y)}$.

$$\begin{aligned}
&\mathsf{del(d,a,0)} = 1\\
&\mathsf{del(d,a,Sy)} = \pi_0^{\mathsf{exp(d,y)}} \star \mathsf{del(d,a,y)} \quad \text{if} \quad \sim(\exists\mathsf{i} < \mathsf{len(a)})[\mathsf{exp(a,i)} = \mathsf{exp(d,y)}]\\
&\mathsf{del(d,a,Sy)} = \mathsf{del(d,a,y)} \quad \text{otherwise}
\end{aligned}$$

So del picks out the members of d that are not in a (since the length of $\mathsf{1}$ is $\mathsf{0}$, $\mathsf{a} \star \mathsf{1}$ is just a). Then $\mathsf{d} \ominus \mathsf{a} =_{\text{def}} \mathsf{del}(\mathsf{d}, \mathsf{a}, \mathsf{len}(\mathsf{d}))$. Now the base cases for the functions are straightforward.

$$\mathsf{space}(\mathsf{m}, \mathsf{n}, 0) = \mathsf{code}(\mathsf{n})$$

$$\mathsf{state}(\mathsf{m}, \mathsf{n}, 0) = \mathsf{machs}(\mathsf{m}, \langle 1 \rangle, \mathsf{space}(\mathsf{m}, \mathsf{n}, 0))$$

And where $\mathsf{state}(\mathsf{m}, \mathsf{n}, \mathsf{j})$ is some $\langle q_i, S_a, S_b, q_j \rangle$ say the *active* area is the n-space of $\mathsf{space}(\mathsf{m}, \mathsf{n}, \mathsf{j})$ where n is the depth of S_a; and for an active area a, the *complement* space is $\mathsf{space}(\mathsf{m}, \mathsf{n}, \mathsf{j}) \ominus \mathsf{a}$. Then,

$\mathsf{space}(\mathsf{m}, \mathsf{n}, \mathsf{Sj}) =$ the least y such that there are maps a on S_a and b on S_b, and
S_a matches the active area on map a, and
a and b agree on the mapping of any cell in the border of S_a, and
b maps any cell not in the border of S_a to a cell not in the complement space, and
y is the projection of b with S_b, concatenated to the complement space.

The idea is to delete the cells from $\mathsf{state}(\mathsf{m}, \mathsf{n}, \mathsf{j})$ that are matched with S_a and replace them with the cells from S_b; for this, it is important to get the mappings to "line up" so that the borders match as they should, and new cells do not walk on old ones; once this is done, the replacement is straightforward. So there is a map a on which S_a matches the active area and a map b that gives the "destination" cells for S_b. Map b is such that: numbers of border cells are properly connected up with the existing dataspace; cells not in the border are sent to open numbers; and the new dataspace consists of the complement space together with the projected cells from b and S_b. Then,

$$\mathsf{state}(\mathsf{m}, \mathsf{n}, \mathsf{Sj}) = \mathsf{machs}(\mathsf{m}, \mathsf{exp}(\mathsf{state}(\mathsf{m}, \mathsf{n}, \mathsf{j}), 3), \mathsf{space}(\mathsf{m}, \mathsf{n}, \mathsf{Sj}))$$

At this stage, functions for $\mathsf{stop}(\mathsf{i}, \mathsf{n}, \mathsf{j})$ and $\mathsf{f}(\mathsf{n})$ are as before.

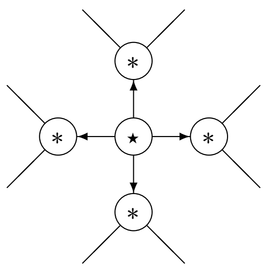
There are a lot of details (and you have a chance to work some of them out in exercises)! But it should already be clear that any K-U computable function is recursive and so that T14.11 is established.

Thus the squeezing argument is complete: Turing computability implies algorithmic computability and algorithmic computability implies K-U computability. But every K-U computable function is recursive and so Turing computable. So the algorithmically computable functions are the same as the Turing computable functions. So Church's thesis!

This argument is just as strong as the premise that algorithmic computability implies K-U computability. For this, we have *translated* an informal notion into

a formal one. Perhaps it is difficult to imagine an algorithmic method that does not conform to AC and then KU. The idea has been to develop the *definition* of an algorithm. Still, this strategy is vulnerable to the charge that we have somehow excluded from the formal account methods that are properly algorithmic, though not Turing computable. There are different responses. First, we should be clear about the range of K-U computability. Say we are interested in parallel computing, whether by individuals following instructions or computing devices. A K-U machine has but a single origin; this might seem to be a problem. Still, an active area might have many "shapes" — and things might be set up as follows,

(P)



with "satellite" centers, to achieve the effect of parallel computing. So it is important to recognize the generality already built into the K-U machine.

Second, it may be that we have ruled out some method that is properly algorithmic, but that our strategies naturally adapt to show that this new method calculates nothing but recursive functions as well. So, for example, cells in our implementation of the K-U machine stand just in binary relations. An obvious extension would be to allow relations other than binary. Given an extended argument to show that the result computes recursive functions, Church's thesis is not threatened.

Finally, it may be that our argument goes some distance to illuminating the *effective range* of the equation between computability and recursive functions. Perhaps the K-U machine is plausible as a technical *specification* for algorithmic computability — or for a specific (and important) sort of algorithmic computability. Then Church's thesis is demonstrably true with respect to it. Perhaps Zeno's machine or the M-H machine computes functions other than recursive functions. Still, insofar as these are not algorithmic (or of the specified sort), they will be irrelevant to the thesis as specified. In this case, Church's thesis is precisely clarified and so established.

To the extent that Church's thesis is either plausible or established, our limiting results become full-fledged *incomputability* results. And, together with incompleteness for our logical systems they are foundational to thinking about the subject matter.

Theorems of chapter 14

T14.1 There is a recursive enumeration of the Turing machines.

T14.2 Every Turing computable function is a recursive function.

T14.3 Every recursive function is Turing computable.

T14.4 There is no Turing machine $\mathsf{H}(\mathsf{i})$ such that $\mathsf{H}(\mathsf{i}) = \mathsf{0}$ if $\Pi_{\mathsf{i}}(\mathsf{i})$ halts and $\mathsf{H}(\mathsf{i}) = \mathsf{1}$ if it does not.

T14.5 There is no Turing machine $\mathsf{H}(\mathsf{i}, \mathsf{n})$ such that $\mathsf{H}(\mathsf{i}, \mathsf{n}) = \mathsf{0}$ if $\Pi_{\mathsf{i}}(\mathsf{n})$ halts and $\mathsf{H}(\mathsf{i}, \mathsf{n}) = \mathsf{1}$ if it does not.

T14.6 If Q is ω-consistent, then no Turing computable function $\mathsf{f}(\mathsf{n})$ is such that $\mathsf{f}(\mathsf{n}) = \mathsf{0}$ just in case n numbers a theorem of predicate logic.

T14.7 The set of truths of $\mathscr{L}_{NT}$ is not recursively enumerable.

T14.8 If T is a recursively axiomatized sound theory whose language includes $\mathscr{L}_{NT}$, then T is negation incomplete.

T14.9 If T is a recursively axiomatized theory extending Q, then there is a sentence $\mathscr{P}$ such that if T is consistent $T \nvdash \mathscr{P}$, and if T is ω-consistent, $T \nvdash \sim\mathscr{P}$.

T14.10 Every (total) function that can be captured by a recursively axiomatized consistent theory extending Q is recursive.

T14.11 Every KU computable function is a recursive function.

And we mention,

CT *Church's Thesis*: The total numerical functions that are effectively computable by some algorithmic method are just the recursive functions.

*E14.14. Assuming functions $\mathsf{code}(\mathsf{n})$ and $\mathsf{decode}(\mathsf{d})$, use the outline in the text to complete the demonstration that any K-U computable function $\mathsf{f}(\mathsf{n})$ is recursive.

E14.15. For each of the following concepts, explain in an essay of about two pages, so that Hannah could understand. In your essay, you should (i) identify the objects to which the concept applies, (ii) give and explain the definition, and give and explicate examples (iii) where the concept applies, and (iv) where it does not. Your essay should exhibit an understanding of methods from the text.

a. The Turing computable functions, and their relation to the recursive functions.

b. The essential elements from the chapter contributing to a demonstration of the decision problem, along with the significance of Church's thesis for this result.

c. The essential elements from this chapter contributing to a demonstration of (the semantic version of) the incompleteness of arithmetic.

d. Church's thesis, along with reasons for thinking it is true, including the possibility of demonstrating its truth.

Concluding Remarks

Looking Forward and Back

We began this text in Part I setting up the elements of classical symbolic logic. Thus we began with four notions of validity: logical validity, validity in our derivation systems *AD* and *ND*, along with semantic (sentential and) quantificational validity. After a parenthesis in Part II to think about techniques for reasoning about logic, we began to put those techniques to work. The main burden of Part III was to show soundness and adequacy of our classical logic, that $\Gamma \vdash \mathcal{P}$ iff $\Gamma \vDash \mathcal{P}$. This is the good news. In Part IV we established some limiting results. These include Gödel's first and second theorems, that no consistent, recursively axiomatizable extension of Q is negation complete, and that no consistent recursively axiomatized theory extending PA proves its own consistency. Results about derivations are associated with computations, and the significance of this association extended by means of Church's thesis. This much constitutes a solid introduction to classical logic, and should position you make progress in logic and philosophy, along with related areas of mathematics and computer science.

Excellent texts which mostly overlap the content of one, but extend it in different ways are Mendelson, *Introduction to Mathematical Logic*; Enderton, *Introduction to Mathematical Logic*; and Boolos, Burgess and Jeffrey, *Computability and Logic*; these put increased demands on the reader (and such demands are one motivation for our text), but should be accessible to you now; Schonfield, *Introduction to Mathematical Logic* is excellent yet still more difficult. Smith, *An Introduction to Gödel's Theorems* extends the material of Part IV. Much of what we have done presumes some set theory as Enderton, *Elements of Set Theory*, or model theory as Manzano, *Model Theory* and, more advanced, Hodges, *A Shorter Model Theory*.

In places, we have touched on logics alternative to classical logic, including multi-valued logic, modal logic, and logics with alternative accounts of the conditional. A good place to start is Priest, *Non-Classical Logics*, which is profitably read with Roy, "Natural Derivations for Priest" which introduces derivations in a style much like our own. Our logic is *first-order* insofar as quantifiers bind just variables

for objects. Second-order logic lets quantifiers bind variables for predicates as well (so $\forall x \forall y[x = y \to \forall F(Fx \leftrightarrow Fy)]$ expresses the *indiscernibility of identicals*). Second-order logic has important applications in mathematics, and raises important issues in metalogic. For this, see Shapiro, *Foundations Without Foundationalism*, and Manzano, *Extensions of First Order Logic*.

Philosophy of logic and mathematics is a subject matter of its own. Shapiro, "Philosophy of Mathematics and Its Logic" (along with the rest of the articles in the *Oxford Handbook*, and Shapiro, *Thinking About Mathematics* are a good place to start. Benacerraf and Putnam, *Philosophy of Mathematics* is a collection of classic articles.

Smith's online, "Teach Yourself Logic" is an excellent comprehensive guide to further resources.

Have fun!

Answers to Selected Exercises

Chapter One

E1.1. Say whether each of the following stories is internally consistent or inconsistent. In either case, explain why.

a. Smoking cigarettes greatly increases the risk of lung cancer, although most people who smoke cigarettes do not get lung cancer.

   Consistent. Even though the risk of cancer goes up with smoking, it may be that most people who smoke do not have cancer. Perhaps 49% of people who smoke get cancer, and 1% of people who do not smoke get cancer. Then smoking greatly increases the risk, even though most people who smoke do not get it.

c. Abortion is always morally wrong, though abortion is morally right in order to save a woman's life.

   Inconsistent. Suppose (whether you believe it or not) that abortion is *always* morally wrong. Then it is wrong to save a woman's life. So the story requires that it is and is not wrong to save a woman's life.

e. No rabbits are nearsighted, though some rabbits wear glasses.

   Consistent. One reason for wearing glasses is to correct nearsightedness. But glasses may be worn for other reasons. It might be that rabbits who wear glasses are farsighted, or have astigmatism, or think that glasses are stylish. Or maybe they wear sunglasses just to look cool.

g. Bill Clinton was never president of the United States, although Hillary is president right now.

   Consistent. Do not get confused by the facts! In a story it may be that Bill was never president and his wife was. Thus this story does not contradict itself and is consistent.

i. The death star is a weapon more powerful than that in any galaxy, though there is, in a galaxy far far away, a weapon more powerful than it.

   Inconsistent. If the death star is more powerful than any weapon in any galaxy, then according to this story it is and is not more powerful than the weapon in the galaxy far far away.

E1.2. For each of the following sentences, (i) say whether it is true or false in the real world and then (ii) say if you can whether it is true or false according to the accompanying story. In each case, explain your answers.

Exercise 1.2

c. Sentence: After overrunning Phoenix in early 2006, a herd of buffalo overran Newark, New Jersey.

   Story: A thundering herd of buffalo overran Phoenix Arizona in early 2006. The city no longer exists.

   (i) It is *false* in the real world that any herd of buffalo overran Newark anytime after 2006. (ii) And, though the story says something about Phoenix, the story does not contain enough information to say whether the sentence regarding Newark is true or false.

e. Sentence: Jack Nicholson has swum the Atlantic.

   Story: No human being has swum the Atlantic. Jack Nicholson and Bill Clinton and you are all human beings, and at least one of you swam all the way across!

   (i) It is *false* in the real world that Jack Nicholson has swum the Atlantic. (ii) This story is inconsistent! It requires that some human both has and has not swum the Atlantic. Thus we refuse to say that it makes the sentence true or false.

g. Sentence: Your instructor is not a human being.

   Story: No beings from other planets have ever made it to this country. However, your instructor made it to this country from another planet.

   (i) Presumably, the claim that your instructor is not a human being is *false* in the real world (assuming that you are not working by independent, or computer-aided study). (ii) But this story is inconsistent! It says both that no beings from other planets have made it to this country and that some being has. Thus we refuse to say that it makes any sentence true or false.

i. Sentence: The Yugo is the most expensive car in the world.

   Story: Jaguar and Rolls Royce are expensive cars. But the Yugo is more expensive than either of them.

   (i) The Yugo is a famously cheap automobile. So the sentence is *false* in the real world. (ii) According to the story, the Yugo is more expensive than some expensive cars. But this is not enough information to say whether it is the most expensive car in the world. So there is not enough information to say whether the sentence is true or false.

Exercise 1.2.i

E1.3. Use our invalidity test to show that each of the following arguments is not logically valid, and so not logically sound.

*For each of these problems, different stories might do the job.

a. If Joe works hard, then he will get an 'A'
   Joe will get an 'A'
   ———
   Joe works hard

   a. In any story with premises true and conclusion false,
      1. If Joe works hard, then he will get an 'A'
      2. Joe will get an 'A'
      3. Joe does not work hard
   b. Story: Joe is very smart, and if he works hard, then he will get an 'A'. Joe will get an 'A'; however, Joe cheats and gets the 'A' without working hard.
   c. This is a consistent story that makes the premises true and the conclusion false; thus, by definition, the argument is not logically valid.
   d. Since the argument is not logically valid, by definition, it is not logically sound.

E1.4. Use our validity procedure to show that each of the following is logically valid, and to decide (if you can) whether it is logically sound.

*For each of these problems, particular reasonings might take different forms.

a. If Bill is president, then Hillary is first lady
   Hillary is not first lady
   ———
   Bill is not president

   a. In any story with premises true and conclusion false,
      (1) If Bill is president, then Hillary is first lady
      (2) Hillary is not first lady
      (3) Bill is president
   b. In any such story,
      Given (1) and (3),
      (4) Hillary is first lady
      Given (2) and (4),
      (5) Hillary is and is not first lady

Exercise 1.4.a

c. So no story with the premises true and conclusion false is a consistent story; so by definition, the argument is logically valid.

d. In the real world Hillary is not first lady and Bill and Hillary are married so it is true that if Bill is president, then Hillary is first lady; so all the premises are true and by definition the argument is logically sound.

E1.5. Use our procedures to say whether the following are logically valid or invalid, and sound or unsound. Hint: You may have to do some experimenting to decide whether the arguments are logically valid or invalid — and so to decide which procedure applies.

c. Some dogs have red hair
Some dogs have long hair
———
Some dogs have long red hair

a. In any story with the premise true and conclusion false,
   1. Some dogs have red hair
   2. Some dogs have long hair
   3. No dogs have long red hair

b. Story: There are dogs with red hair, and there are dogs with long hair. However, due to a genetic defect, no dogs have long red hair.

c. This is a consistent story that makes the premise true and the conclusion false; thus, by definition, the argument is not logically valid.

d. Since the argument is not logically valid, by definition, it is not logically sound.

E1.6. Use our procedures to say whether the following are logically valid or invalid, and sound or unsound.

c. The earth is (approximately) round
———
There is no round square

a. In any story with the premise true and conclusion false,
   (1) The earth is (approximately) round
   (2) There is a round square

b. In any such story, given (2),
   (3) Something is round and not round

Exercise 1.6.c

c. So no story with the premises true and conclusion false is a consistent story; so by definition, the argument is logically valid.

d. In the real world the earth is (approximately) round, so the premise is true and by definition the argument is logically sound.

E1.8. Which of the following are true, and which are false? In each case, explain your answers, with reference to the relevant definitions.

c. If the conclusion of an argument is true in the real world, then the argument must be logically valid.

False. An argument is logically valid iff there is no consistent story that makes the premises true and the conclusion false. Though the conclusion is true in the real world (and so in the true story), there may be some other story that makes the premises true and the conclusion false. If this is so, then the argument is not logically valid.

e. If a premise of an argument is false in the real world, then the argument cannot be logically valid.

False. An argument is logically valid iff there is no consistent story that makes the premises true and the conclusion false. For logical validity, there is no requirement that every story have true premises — only that ones that do, also have true conclusions. So an argument might be logically valid, and have premises that are false in many stories, including the true story.

g. If an argument is logically sound, then its conclusion is true in the real world.

True. An argument is logically valid iff there is no consistent story that makes the premises true and the conclusion false. An argument is logically sound iff it is logically valid and its premises are true in the real world. Since the premises are true in the real world, they hold in the true story; since the argument is valid, this story cannot be one where the conclusion is false. So the conclusion of a sound argument is true in the real world.

i. If the conclusion of an argument cannot be false (is false in no consistent story), then the argument is logically valid.

True. If there is no consistent story where the conclusion is false, then there is no consistent story where the premises are true and the conclusion is false; but an argument is logically valid iff there is no consistent story where the premises are true and the conclusion is false. So the argument is logically valid.

Exercise 1.8.i

Chapter Two

E2.1. Assuming that $\mathscr{S}$ may represent any sentence letter, and $\mathscr{P}$ any arbitrary expression of $\mathcal{L}_s$, use maps to determine whether each of the following expressions is (i) of the form $(\mathscr{S} \to \sim\mathscr{P})$ and then (ii) whether it is of the form $(\mathscr{P} \to \sim\mathscr{P})$. In each case, explain your answers.

e. $((\to \sim) \to \sim(\to \sim))$

$$(\mathscr{S} \to \sim\mathscr{P}) \qquad\qquad (\mathscr{P} \to \sim\mathscr{P})$$

$$(\overbrace{(\to \sim)}^{?} \to \sim \overbrace{(\to \sim)}) \qquad\qquad (\overbrace{(\to \sim)} \to \sim \overbrace{(\to \sim)})$$

(i) Since $(\to \sim)$ is not a sentence letter, there is nothing to which $\mathscr{S}$ maps, and $((\to \sim) \to \sim(\to \sim))$ is not of the form $(\mathscr{S} \to \sim\mathscr{P})$. (ii) Since $\mathscr{P}$ maps to any expression, $((\to \sim) \to \sim(\to \sim))$ is of the form $(\mathscr{P} \to \sim\mathscr{P})$ by the above map.

E2.3. For each of the following expressions, demonstrate that it is a formula and a sentence of $\mathcal{L}_s$ with a tree. Then on the tree (i) bracket all the subformulas, (ii) box the immediate subformula(s), (iii) star the atomic subformulas, and (iv) circle the main operator.

a. A

subformula: [$A^{\star}$ This is a formula by FR(s)

In this case, the "tree" is very simple. There are no operators, and so no main operator. There are no immediate subformulas.

E2.4. Explain why the following expressions are not formulas or sentences of $\mathcal{L}_s$. Hint: you may find that an attempted tree will help you see what is wrong.

b. $(\mathscr{P} \to \mathscr{Q})$

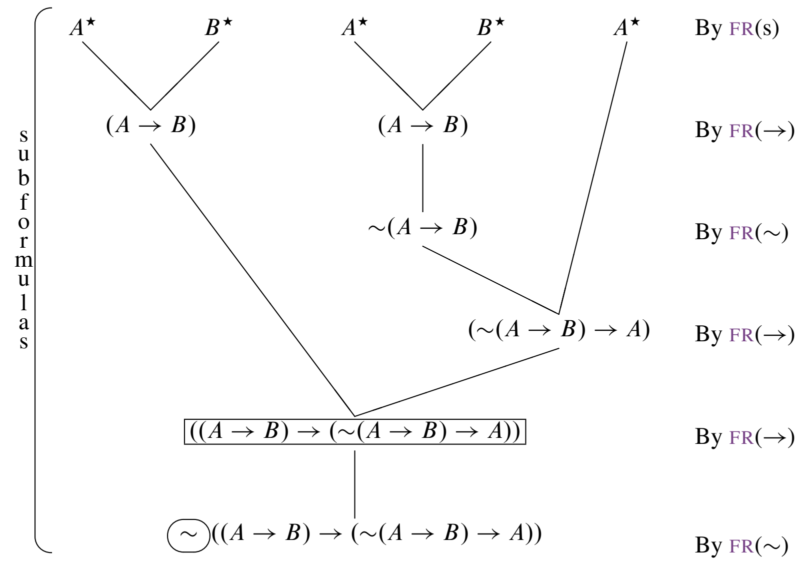
This is not a formula because $\mathscr{P}$ and $\mathscr{Q}$ are not sentence letters of $\mathcal{L}_s$. They are part of the metalanguage by which we describe $\mathcal{L}_s$, but are not among the Roman italics (with or without subscripts) that are the sentence letters. Since it is not a formula, it is not a sentence.

E2.5. For each of the following expressions, determine whether it is a formula and sentence of $\mathcal{L}_s$. If it is, show it on a tree, and exhibit its parts as in E2.3. If it is not, explain why as in E2.4.

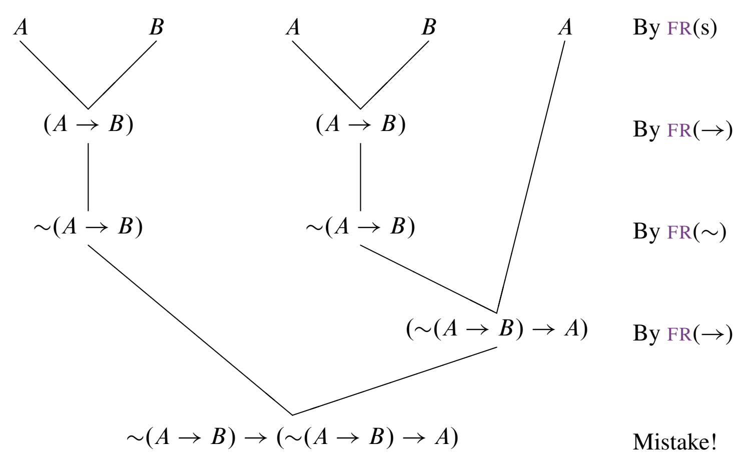
Exercise 2.5

a. $\sim((A \to B) \to (\sim(A \to B) \to A))$

This is a formula and a sentence.



c. $\sim(A \to B) \to (\sim(A \to B) \to A)$

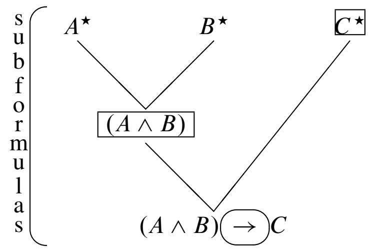


Not a formula or sentence. The attempt to apply FR($\to$) at the last step fails, insofar as the outer parentheses are missing.

E2.6. For each of the following expressions, demonstrate that it is a formula and a sentence of $\mathcal{L}_s$ with a tree. Then on the tree (i) bracket all the subformulas, (ii) box the immediate subformula(s), (iii) star the atomic subformulas, and (iv) circle the main operator.

Exercise 2.6

a. $(A \wedge B) \rightarrow C$



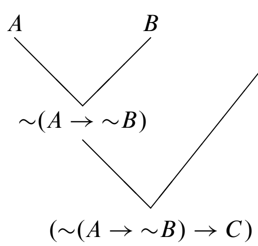
Formulas by FR(s)

Formula by FR$'$($\wedge$)

Formula by FR($\rightarrow$), outer parentheses dropped

E2.7. For each of the formulas in E2.6a - e, produce an unabbreviating tree to find the unabbreviated expression it represents.

a. $(A \wedge B) \rightarrow C$

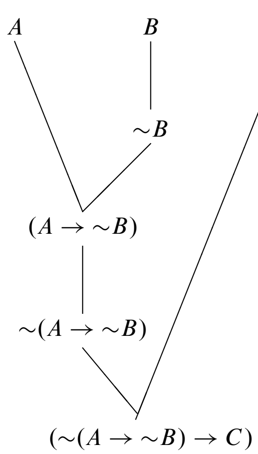


By AB($\wedge$)

Adding outer ( )

E2.8. For each of the unabbreviated expressions from E2.7a - e, produce a complete tree to show by direct application FR that it is an official formula.

a. $(\sim(A \rightarrow \sim B) \rightarrow C)$



Formulas by FR(s)

Formula by FR($\sim$)

Formula by FR($\rightarrow$)

Formula by FR($\sim$)

Formula by FR($\rightarrow$)

Chapter Three

E3.1. Where *A1* is as above, construct derivations to show each of the following.

a. $\mathscr{A} \wedge (\mathscr{B} \wedge \mathscr{C}) \vdash_{A1} \mathscr{B}$

1.	$\mathscr{A} \wedge (\mathscr{B} \wedge \mathscr{C})$	prem
2.	$[\mathscr{A} \wedge (\mathscr{B} \wedge \mathscr{C})] \to (\mathscr{B} \wedge \mathscr{C})$	$\wedge 2$
3.	$\mathscr{B} \wedge \mathscr{C}$	2,1 MP
4.	$(\mathscr{B} \wedge \mathscr{C}) \to \mathscr{B}$	$\wedge 1$
5.	$\mathscr{B}$	4,3 MP

E3.2. Provide derivations for T3.6, T3.7, T3.9, T3.10, T3.11, T3.12, T3.13, T3.14, T3.15, T3.16, T3.18, T3.19, T3.20, T3.21, T3.22, T3.23, T3.24, T3.25, and T3.26. As you are working these problems, you may find it helpful to refer to the *AD* summary on p. 87.

T3.12. $\vdash_{AD} (\mathscr{A} \to \mathscr{B}) \to ({\sim}{\sim}\mathscr{A} \to {\sim}{\sim}\mathscr{B})$

1.	${\sim}{\sim}\mathscr{A} \to \mathscr{A}$	T3.10
2.	$({\sim}{\sim}\mathscr{A} \to \mathscr{A}) \to [(\mathscr{A} \to \mathscr{B}) \to ({\sim}{\sim}\mathscr{A} \to \mathscr{B})]$	T3.5
3.	$(\mathscr{A} \to \mathscr{B}) \to ({\sim}{\sim}\mathscr{A} \to \mathscr{B})$	2,1 MP
4.	$\mathscr{B} \to {\sim}{\sim}\mathscr{B}$	T3.11
5.	$(\mathscr{B} \to {\sim}{\sim}\mathscr{B}) \to [({\sim}{\sim}\mathscr{A} \to \mathscr{B}) \to ({\sim}{\sim}\mathscr{A} \to {\sim}{\sim}\mathscr{B})]$	T3.4
6.	$({\sim}{\sim}\mathscr{A} \to \mathscr{B}) \to ({\sim}{\sim}\mathscr{A} \to {\sim}{\sim}\mathscr{B})$	5,4 MP
7.	$(\mathscr{A} \to \mathscr{B}) \to ({\sim}{\sim}\mathscr{A} \to {\sim}{\sim}\mathscr{B})$	3,6 T3.2

T3.16. $\vdash_{AD} \mathscr{A} \to [{\sim}\mathscr{B} \to {\sim}(\mathscr{A} \to \mathscr{B})]$

1.	$(\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{B})$	T3.1
2.	$\mathscr{A} \to [(\mathscr{A} \to \mathscr{B}) \to \mathscr{B}]$	1 T3.3
3.	$[(\mathscr{A} \to \mathscr{B}) \to \mathscr{B}] \to [{\sim}\mathscr{B} \to {\sim}(\mathscr{A} \to \mathscr{B})]$	T3.13
4.	$\mathscr{A} \to [{\sim}\mathscr{B} \to {\sim}(\mathscr{A} \to \mathscr{B})]$	2,3 T3.2

T3.21. $\mathscr{A} \to (\mathscr{B} \to \mathscr{C}) \vdash_{AD} (\mathscr{A} \wedge \mathscr{B}) \to \mathscr{C}$

1.	$\mathscr{A} \to (\mathscr{B} \to \mathscr{C})$	prem
2.	$(\mathscr{B} \to \mathscr{C}) \to ({\sim}\mathscr{C} \to {\sim}\mathscr{B})$	T3.13
3.	$\mathscr{A} \to ({\sim}\mathscr{C} \to {\sim}\mathscr{B})$	1,2 T3.2
4.	${\sim}\mathscr{C} \to (\mathscr{A} \to {\sim}\mathscr{B})$	3, T3.3
5.	$[{\sim}\mathscr{C} \to (\mathscr{A} \to {\sim}\mathscr{B})] \to [{\sim}(\mathscr{A} \to {\sim}\mathscr{B}) \to \mathscr{C}]$	T3.14
6.	${\sim}(\mathscr{A} \to {\sim}\mathscr{B}) \to \mathscr{C}$	5,4 MP
7.	$(\mathscr{A} \wedge \mathscr{B}) \to \mathscr{C}$	6 abv

E3.3. For each of the following, expand the derivations to include all the steps from theorems. The result should be a derivation in which each step is either a premise, an axiom, or follows from previous lines by a rule.

b. Expand the derivation for T3.4

1. $(\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]$ A1
2. $[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$ A2
3. $([\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]) \to$
   $[(\mathcal{B} \to \mathcal{C}) \to ([\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})])]$ A1
4. $(\mathcal{B} \to \mathcal{C}) \to ([\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})])$ 3,2 MP
5. $[(\mathcal{B} \to \mathcal{C}) \to ([\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})])] \to$
   $[((\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]) \to ((\mathcal{B} \to \mathcal{C}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})])]$ A2
6. $((\mathcal{B} \to \mathcal{C}) \to [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})]) \to ((\mathcal{B} \to \mathcal{C}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})])$ 5,4 MP
7. $(\mathcal{B} \to \mathcal{C}) \to [(\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to \mathcal{C})]$ 6,1 MP

E3.4. Consider an axiomatic system *A2* as described in the main problem. Provide derivations for each of the following, where derivations may appeal to any *prior* result (no matter what *you* have done).

a. $\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C} \vdash_{A2} \sim(\sim\mathcal{C} \wedge \mathcal{A})$

1. $\mathcal{A} \to \mathcal{B}$ prem
2. $(\mathcal{A} \to \mathcal{B}) \to [\sim(\mathcal{B} \wedge \sim\mathcal{C}) \to \sim(\sim\mathcal{C} \wedge \mathcal{A})]$ A3
3. $\sim(\mathcal{B} \wedge \sim\mathcal{C}) \to \sim(\sim\mathcal{C} \wedge \mathcal{A})$ 2,1 MP
4. $\mathcal{B} \to \mathcal{C}$ prem
5. $\sim(\mathcal{B} \wedge \sim\mathcal{C})$ 4 abv
6. $\sim(\sim\mathcal{C} \wedge \mathcal{A})$ 5,3 MP

d. $\vdash_{A2} \sim(\mathcal{A} \wedge \mathcal{B}) \to (\mathcal{B} \to \sim\mathcal{A})$

1. $\sim\sim\mathcal{A} \to \mathcal{A}$ (c)
2. $(\sim\sim\mathcal{A} \to \mathcal{A}) \to [\sim(\mathcal{A} \wedge \mathcal{B}) \to \sim(\mathcal{B} \wedge \sim\sim\mathcal{A})]$ A3
3. $\sim(\mathcal{A} \wedge \mathcal{B}) \to \sim(\mathcal{B} \wedge \sim\sim\mathcal{A})$ 2,1 MP
4. $\sim(\mathcal{A} \wedge \mathcal{B}) \to (\mathcal{B} \to \sim\mathcal{A})$ 3 abv

g. $\sim\mathcal{A} \to \sim\mathcal{B} \vdash_{A2} \mathcal{B} \to \mathcal{A}$

1. $\sim\mathcal{A} \to \sim\mathcal{B}$ prem
2. $(\sim\mathcal{A} \to \sim\mathcal{B}) \to [\sim(\sim\mathcal{B} \wedge \mathcal{B}) \to \sim(\mathcal{B} \wedge \sim\mathcal{A})]$ A3
3. $\sim(\sim\mathcal{B} \wedge \mathcal{B}) \to \sim(\mathcal{B} \wedge \sim\mathcal{A})$ 2,1 MP
4. $\sim(\sim\mathcal{B} \wedge \mathcal{B})$ (b)
5. $\sim(\mathcal{B} \wedge \sim\mathcal{A})$ 3,4 MP
6. $\mathcal{B} \to \mathcal{A}$ 5 abv

Exercise 3.4.g

i. $\mathcal{A} \to \mathcal{B}, \mathcal{B} \to \mathcal{C}, \mathcal{C} \to \mathcal{D} \vdash_{A2} \mathcal{A} \to \mathcal{D}$

1.	$\mathcal{A} \to \mathcal{B}$	prem
2.	$\mathcal{B} \to \mathcal{C}$	prem
3.	$\sim(\sim\mathcal{C} \wedge \mathcal{A})$	1,2 (a)
4.	$\mathcal{C} \to \mathcal{D}$	prem
5.	$(\mathcal{C} \to \mathcal{D}) \to (\sim\mathcal{D} \to \sim\mathcal{C})$	(f)
6.	$\sim\mathcal{D} \to \sim\mathcal{C}$	5,4 MP
7.	$(\sim\mathcal{D} \to \sim\mathcal{C}) \to [\sim(\sim\mathcal{C} \wedge \mathcal{A}) \to \sim(\mathcal{A} \wedge \sim\mathcal{D})]$	A3
8.	$\sim(\sim\mathcal{C} \wedge \mathcal{A}) \to \sim(\mathcal{A} \wedge \sim\mathcal{D})$	7,6 MP
9.	$\sim(\mathcal{A} \wedge \sim\mathcal{D})$	8,3 MP
10.	$\mathcal{A} \to \mathcal{D}$	9 abv

u. $\vdash_{A2} [\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C}]$

1.	$[(\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C}] \to [\mathcal{A} \wedge (\mathcal{B} \wedge \sim\mathcal{C})]$	(s)
2.	$(\mathcal{B} \wedge \sim\mathcal{C}) \to \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})$	(e)
3.	$[\mathcal{A} \wedge (\mathcal{B} \wedge \sim\mathcal{C})] \to [\mathcal{A} \wedge \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})]$	2 (q)
4.	$[(\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C}] \to [\mathcal{A} \wedge \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})]$	1,3 (l)
5.	$([(\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C}] \to [\mathcal{A} \wedge \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})]) \to$ $(\sim[\mathcal{A} \wedge \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})] \to \sim[(\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C}])$	(f)
6.	$\sim[\mathcal{A} \wedge \sim\sim(\mathcal{B} \wedge \sim\mathcal{C})] \to \sim[(\mathcal{A} \wedge \mathcal{B}) \wedge \sim\mathcal{C}]$	5,4 MP
7.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C}]$	6 abv

w. $\mathcal{A} \to \mathcal{B}, \mathcal{A} \to (\mathcal{B} \to \mathcal{C}) \vdash_{A2} \mathcal{A} \to \mathcal{C}$

1.	$\mathcal{A} \to (\mathcal{B} \to \mathcal{C})$	prem
2.	$[\mathcal{A} \to (\mathcal{B} \to \mathcal{C})] \to [(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C}]$	(u)
3.	$(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{C}$	2,1 MP
4.	$\mathcal{A} \to \mathcal{A}$	(j)
5.	$\mathcal{A} \to \mathcal{B}$	prem
6.	$\mathcal{A} \to (\mathcal{A} \wedge \mathcal{B})$	4,5 (r)
7.	$\mathcal{A} \to \mathcal{C}$	6,3 (l)

Chapter Four

E4.1. Where the interpretation is as in J from p. 97, use trees to decide whether the following sentences of $\mathcal{L}_s$ are T or F.

a. $\sim A$ *false*

$A^{(\mathsf{T})}$ From J

|

$\sim A^{(\mathsf{F})}$ By T($\sim$), row 1

Exercise 4.1.a

e. $\sim(A \to A)$ *false*

$A^{(\mathsf{T})}$ $\quad A^{(\mathsf{T})}$ From J

$(A \to A)^{(\mathsf{T})}$ By T(→), row 1

$\sim(A \to A)^{(\mathsf{F})}$ By T(∼), row 1

f. $(\sim A \to A)$ *true*

$A^{(\mathsf{T})}$ $\quad A^{(\mathsf{T})}$ From J

$\sim A^{(\mathsf{F})}$ By T(∼), row 1

$(\sim A \to A)^{(\mathsf{T})}$ By T(→), row 3

i. $(A \to \sim B) \to \sim(B \to \sim A)$ *true*

$A^{(\mathsf{T})}$ $\quad B^{(\mathsf{T})}$ $\quad B^{(\mathsf{T})}$ $\quad A^{(\mathsf{T})}$ From J

$\sim B^{(\mathsf{F})}$ $\quad \sim A^{(\mathsf{F})}$ By T(∼), row 1

$(A \to \sim B)^{(\mathsf{F})}$ $\quad (B \to \sim A)^{(\mathsf{F})}$ By T(→), row 2

$\sim(B \to \sim A)^{(\mathsf{T})}$ By T(∼), row 2

$(A \to \sim B) \to \sim(B \to \sim A)^{(\mathsf{T})}$ By T(→), row 3

E4.2. For each of the following sentences of $\mathcal{L}_s$ construct a truth table to determine its truth value for each of the possible interpretations of its basic sentences.

a. $\sim\sim A$

A	$\sim\sim A$
T	***T*** F
F	***F*** T

Exercise 4.2.a

d. $(\sim B \rightarrow A) \rightarrow B$

A	B	$(\sim$	B	$\rightarrow$	$A)$	$\rightarrow$	B
T	T	F		T		***T***	
T	F	T		T		***F***	
F	T	F		T		***T***	
F	F	T		F		***T***	

g. $C \rightarrow (A \rightarrow B)$

A	B	C	C	$\rightarrow$	$(A$	$\rightarrow$	$B)$
T	T	T		***T***		T	
T	T	F		***T***		T	
T	F	T		***F***		F	
T	F	F		***T***		F	
F	T	T		***T***		T	
F	T	F		***T***		T	
F	F	T		***T***		T	
F	F	F		***T***		T	

i. $(\sim A \rightarrow B) \rightarrow (\sim C \rightarrow D)$

A	B	C	D	$(\sim$	A	$\rightarrow$	$B)$	$\rightarrow$	$(\sim$	C	$\rightarrow$	$D)$
T	T	T	T	F		T		***T***	F		T	
T	T	T	F	F		T		***T***	F		T	
T	T	F	T	F		T		***T***	T		T	
T	T	F	F	F		T		***F***	T		F	
T	F	T	T	F		T		***T***	F		T	
T	F	T	F	F		T		***T***	F		T	
T	F	F	T	F		T		***T***	T		T	
T	F	F	F	F		T		***F***	T		F	
F	T	T	T	T		T		***T***	F		T	
F	T	T	F	T		T		***T***	F		T	
F	T	F	T	T		T		***T***	T		T	
F	T	F	F	T		T		***F***	T		F	
F	F	T	T	T		F		***T***	F		T	
F	F	T	F	T		F		***T***	F		T	
F	F	F	T	T		F		***T***	T		T	
F	F	F	F	T		F		***T***	T		F	

E4.3. For each of the following, use truth tables to decide whether the entialment claims hold.

a. $A \rightarrow \sim A \models_s \sim A$ *valid*

A	A	$\rightarrow$	$\sim$	A	/	$\sim$	A
T	T	***F***	F	T		***F***	T
F	F	***T***	T	F		***T***	F

Exercise 4.3.a

c. $A \to B, \sim A \vDash_s \sim B$ *invalid*

A	B	$A \to B$	$\sim A$	/ $\sim B$	
T	T	***T***	***F***	***F***	
T	F	***F***	***F***	***T***	
F	T	***T***	***T***	***F***	$\Leftarrow$
F	F	***T***	***T***	***T***	

g. $\vDash_s [A \to (C \to B)] \to [(A \to C) \to (A \to B)]$ *valid*

A	B	C	$[A \to$	$(C \to B)]$	$\to$	$[(A \to C)$	$\to$	$(A \to B)]$
T	T	T	T	T	***T***	T	T	T
T	T	F	T	T	***T***	F	T	T
T	F	T	F	F	***T***	T	F	F
T	F	F	T	T	***T***	F	T	F
F	T	T	T	T	***T***	T	T	T
F	T	F	T	T	***T***	T	T	T
F	F	T	T	F	***T***	T	T	T
F	F	F	T	T	***T***	T	T	T

E4.4. For each of the following, use truth tables to decide whether the entialment claims hold.

c. $B \vee \sim C \vDash_s B \to C$ *invalid*

B	C	$B \vee$	$\sim C$	/ $B \to C$	
T	T	***T***	F	***T***	
T	F	***T***	T	***F***	$\Leftarrow$
F	T	***F***	F	***T***	
F	F	***T***	T	***T***	

d. $A \vee B, \sim C \to \sim A, \sim(B \wedge \sim C) \vDash_s C$ *valid*

A	B	C	$A \vee B$	$\sim C$	$\to$	$\sim A$	$\sim(B$	$\wedge$	$\sim C)$	/ C
T	T	T	***T***	F	***T***	F	***T***	F	F	***T***
T	T	F	***T***	T	***F***	F	***F***	T	T	***F***
T	F	T	***T***	F	***T***	F	***T***	F	F	***T***
T	F	F	***T***	T	***F***	F	***T***	F	T	***F***
F	T	T	***T***	F	***T***	T	***T***	F	F	***T***
F	T	F	***T***	T	***T***	T	***F***	T	T	***F***
F	F	T	***F***	F	***T***	T	***T***	F	F	***T***
F	F	F	***F***	T	***T***	T	***T***	F	T	***F***

h. $\models_s \sim(A \leftrightarrow B) \leftrightarrow (A \wedge \sim B)$ *invalid*

A	B	$\sim$	$(A$	$\leftrightarrow$	$B)$	$\leftrightarrow$	$(A$	$\wedge$	$\sim$	$B)$	
T	T	F		T		***T***		F	F		
T	F	T		F		***T***		T	T		
F	T	T		F		***F***		F	F		⇐
F	F	F		T		***T***		F	T		

E4.5. For each of the following, use truth tables to decide whether the entailment claims hold.

a. $\exists xAx \rightarrow \exists xBx, \sim\exists xAx \models_s \exists xBx$ *invalid*

$\exists xAx$	$\exists xBx$	$\exists xAx \rightarrow \exists xBx$	$\sim\exists xAx$	/ $\exists xBx$	
T	T	***T***	***F***	***T***	
T	F	***F***	***F***	***F***	
F	T	***T***	***T***	***T***	
F	F	***T***	***T***	***F***	⇐

Chapter Five

E5.1. For each of the following, identify the simple sentences that are parts. If the sentence is compound, use underlines to exhibit its operator structure, and say what is its main operator.

h. Hermoine believes that studying is good, and Hermione studies hard, but Ron believes studying is good, and it is not the case that Ron studies hard.

Simple sentences:
Stydying is good
Hermione studies hard
Ron studies hard

Hermoine believes that studying is good and Hermione studies hard but Ron believes studying is good and it is not the case that Ron studies hard.

main operator: ____ but ____

E5.2. Which of the following operators are truth-functional and which are not? If the operator is truth-functional, display the relevant table; if it is not, give a case to show that it is not. Clearly explain your response.

a. It is a fact that ____ *truth functional*

It is a fact that	____
T	T
F	F

Exercise 5.2.a

In any situation, the compound takes the same value as the sentence in the blank. So the operator is truth-functional.

c. ____ but ____ *truth functional*

____	but	____
T	***T***	T
T	***F***	F
F	***F***	T
F	***F***	F

In any situation this operator takes the same value as ____ and ____. Though 'but' may carry a conversational sense of opposition not present with 'and' the *truth value* of the compound works the same. Thus, where Bob loves Sue even 'Bob loves Sue but Bob loves Sue' might elicit the response "*True*, but why did you say that?"

f. It is always the case that ____ *not truth functional*

It may be that any false sentence in the blank results in a false compound. However, consider something true in the blank: perhaps 'I am at my desk' and 'Life is hard' are both true. But

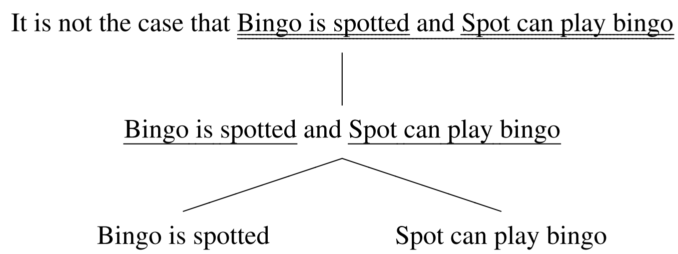
> It is always the case that I am at my desk
>
> It is always the case that life is hard

are such that the first is false, but the second remains true. For perhaps I sometimes get up from my desk (so that the first is false), but the difficult character of living goes on and on (and on). Thus there are situations where truth values of sentences in the blanks are the same, but the truth values of resultant compounds are different. So the operator is not truth-functional.

E5.3. Use our method to expose truth functional structure and produce parse trees for each of the following. Use your trees to produce an interpretation function for the sentences.

d. It is not the case that: Bingo is spotted and Spot can play bingo.

Exercise 5.3.d



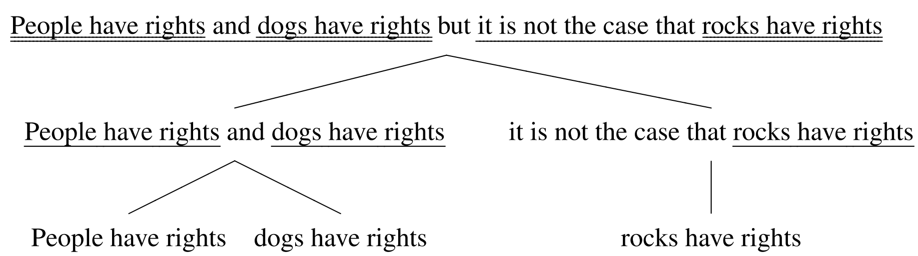
From this sentence, II includes,

B: Bingo is spotted

S: Spot can play bingo

E5.4. Use our method to expose truth functional structure and produce parse trees for each of the following. Use your trees to produce an interpretation function for the sentences.

a. People have rights and dogs have rights, but rocks do not.



From this sentence, II includes,

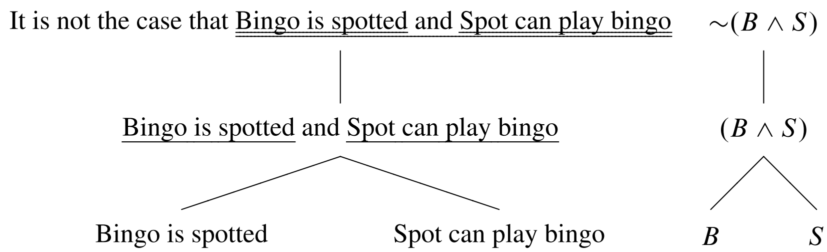
P: People have rights

D: Dogs have rights

R: Rocks have rights

E5.5. Construct parallel trees to complete the translation of the sentences from E5.3 and E5.4.

d. It is not the case that: Bingo is spotted and Spot can play bingo.

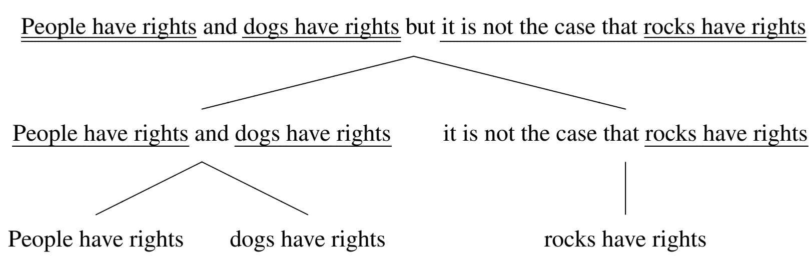


Where II includes,

B: Bingo is spotted

S: Spot can play bingo

a. People have rights and dogs have rights, but rocks do not.

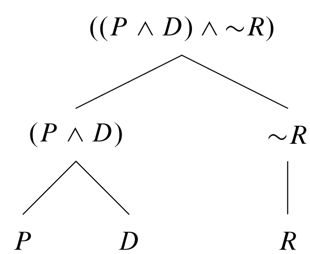


Where II includes,

P: People have rights

D: Dogs have rights

R: Rocks have rights



E5.6. Use our method to translate each of the following. That is, generate parse trees with an interpretation function for all the sentences, and then parallel trees to produce formal equivalents.

c. It is not the case that: everything Plato, and Aristotle, and Ayn Rand said was true.

It is not the case that everything Plato said was true and everything Aristotle said was true and everything Ayn Rand said was true

everything Plato said was true and everything Aristotle said was true and everything Ayn Rand said was true

everything Plato said was true | everything Aristotle said was true and everything Ayn Rand said was true

everything Aristotle said was true | everything Ayn Rand said was true

Where || includes,

P: Everything Plato said was true

A: Everything Aristotle said was true

R: Everything Ayn Rand said was true

$\sim(P \wedge (A \wedge R))$

$P \wedge (A \wedge R))$

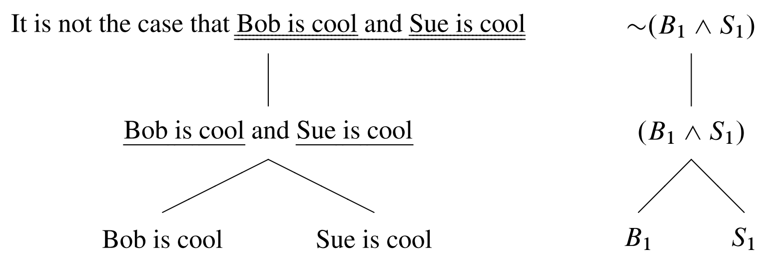
P | $(A \wedge R)$

A | R

Another natural result is
$\sim((P \wedge A) \wedge R)$

E5.8. Using the given interpretation function, produce parse trees and then parallel ones to complete the translation for each of the following.

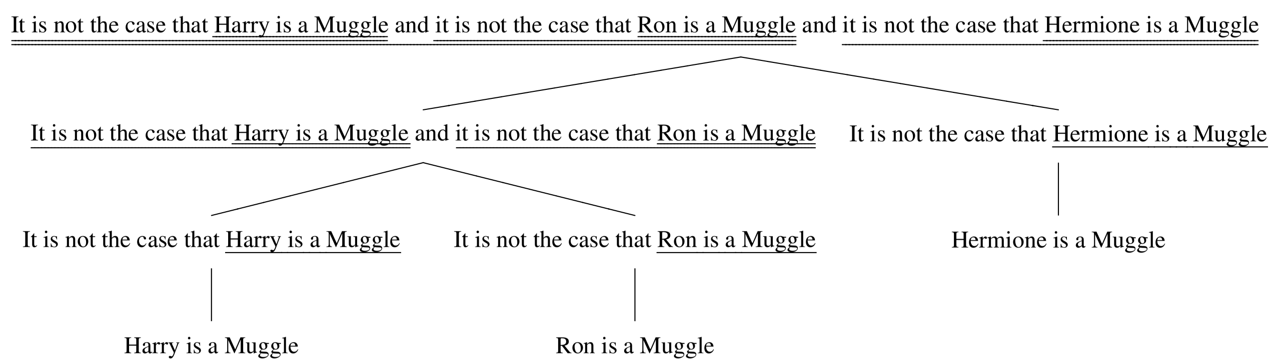
h. Not both Bob and Sue are cool.



E5.9. Use our method to translate each of the following. That is, generate parse trees with an interpretation function for all the sentences, and then parallel trees to produce formal equivalents.

Exercise 5.9

d. Neither Harry, nor Ron, nor Hermione are Muggles.

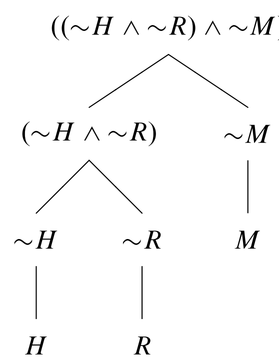


Include in the interpretation function,

H: Harry is a Muggle

R: Ron is a Muggle

M: Hermione is a Muggle



Other natural options are,
$(\sim H \wedge (\sim R \wedge \sim M))$
$\sim((H \vee R) \vee M)$
$\sim(H \vee (R \vee M))$

g. Although blatching and blagging are illegal in Quidditch, the woolongong shimmy is not.

Although It is not the case that blatching is legal in Quidditch and it is not the case that blagging is legal in Quidditch, it is not the case that it is not the case that the woolongon shimmy is legal in Quidditch

It is not the case that blatching is legal in Quidditch and it is not the case that blagging is legal in Quidditch

It is not the case that blatching is legal in Quidditch

blatching is legal in Quidditch

It is not the case that blagging is legal in Quidditch

blagging is legal in Quidditch

it is not the case that it is not the case that the woolongon shimmy is legal in Quidditch

It is not the case that the woolongon shimmy is legal in Quidditch

the woolongon shimmy is legal in Quidditch

Include in the interpretation function,

T: Blatching is legal in Quidditch

G: Blagging is legal in Quidditch

W: The woolongon shimmy is legal in Quidditch

$(\sim T \wedge \sim G) \wedge \sim\sim W)$

$(\sim T \wedge \sim G)$

$\sim T$

T

$\sim G$

G

$\sim\sim W$

$\sim W$

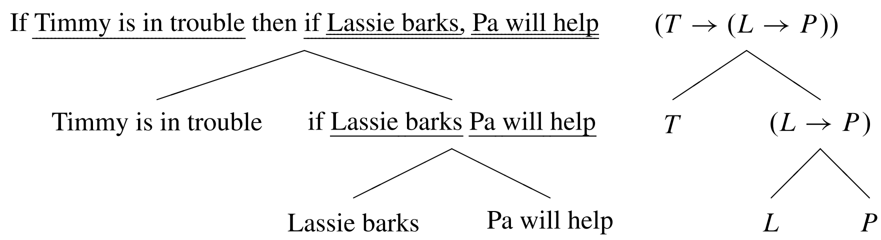
W

It is tempting to build 'Blatching is illegal' and so forth into the interpretation function. But this is to leave out sentential structure. And it is tempting to leave out the double negation on the right conjunct — the result is in fact equivalent, though the above translation reflects the actual claim.

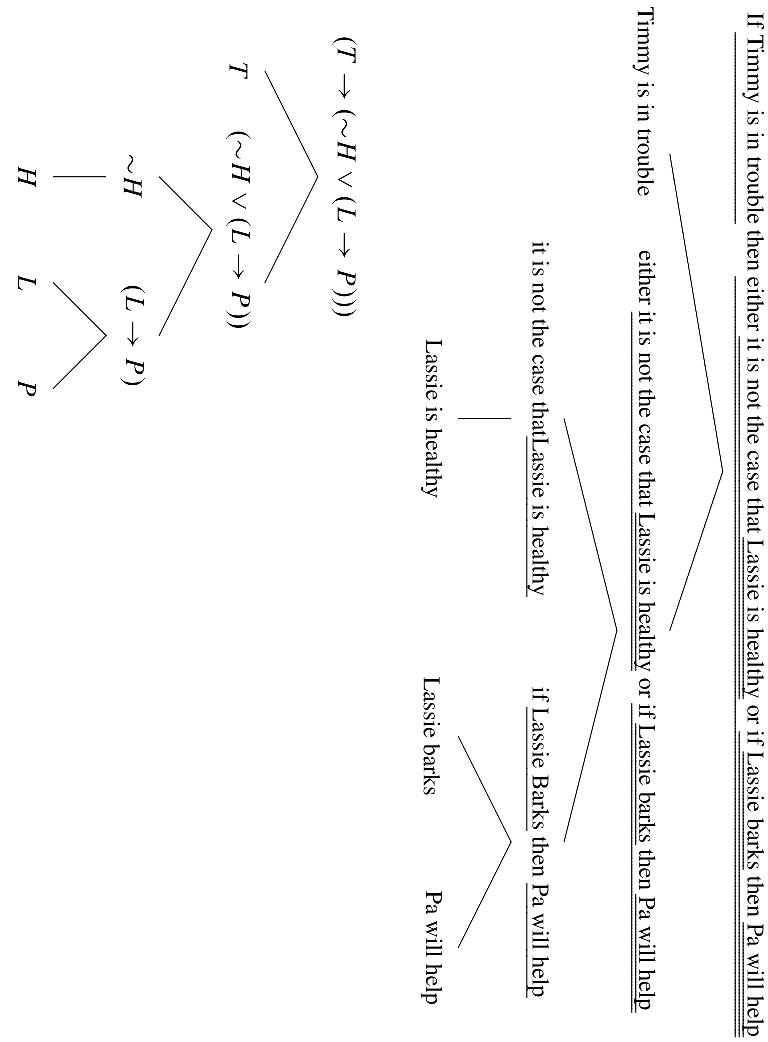
Exercise 5.9.g

E5.10. Using the given interpretation function, produce parse trees and then parallel ones to complete the translation for each of the following.

e. If Timmy is in trouble, then if Lassie barks Pa will help.



i. If Timmy is in trouble, then either Lassie is not healthy or if Lassie barks then Pa will help.



E5.11. Use our method, with or without parse trees, to produce a translation, including interpretation function for the following.

g. If you think animals do not feel pain, then vegetarianism is not right.

Include in the interpretation function,

V: Vegetarianism is right

Exercise 5.11.g

N: You think it is not the case that animals feel pain

$(N \to \sim V)$

i. Vegetarianism is right only if both animals feel pain, and animals have intrinsic value just in case they feel pain; but it is not the case that animals have intrinsic value just in case they feel pain.

Include in the interpretation function,

V: Vegetarianism is right

P: Animals feel pain

I: Animals have intrinsic value

$[V \to (P \wedge (I \leftrightarrow P))] \wedge (\sim I \leftrightarrow P)$

E5.12. For each of the following arguments: (i) Produce an adequate translation, including interpretation function and translations for the premises and conclusion. Then (ii) use truth tables to determine whether the argument is sententially valid.

a. Our car will not run unless it has gasoline
Our car has gasoline

Our car will run

Include in the interpretation function:

R: Our car will run

G: Our car has gasoline

Formal sentences:

$\sim R \vee G$
G

R

Truth table:

G	R	$\sim R$	$\vee$ G	G	/ R	
T	T	F	***T***	***T***	***T***	
T	F	T	***T***	***T***	***F***	⇐
F	T	F	***F***	***F***	***T***	
F	F	T	***T***	***F***	***F***	

Not sententially valid

Exercise 5.12.a

Chapter Six

E6.1. Show that each of the following is valid in N1. Complete (a) - (d) using just rules R1, R3 and R4. You will need an application of R2 for (e).

a. $(A \wedge B) \wedge C \vdash_{N1} A$

1.	$(A \wedge B) \wedge C$	P
2.	$A \wedge B$	1 R3
3.	A	2 R3 Win!

E6.2. (i) For each of the arguments in E6.1, use a truth table to decide if the argument is sententially valid.

a. $(A \wedge B) \wedge C \vdash_{N1} A$

A	B	C	$(A \wedge B) \wedge C$		/	A
T	T	T	T	***T***		T
T	T	F	T	***F***		T
T	F	T	F	***F***		T
T	F	F	F	***F***		T
F	T	T	F	***F***		F
F	T	F	F	***F***		F
F	F	T	F	***F***		F
F	F	F	F	***F***		F

There is no row where the premise is true and the conclusion is false; so this argument is *sententially valid*.

E6.3. Consider a derivation with structure as in the main problem. For each of the lines (3), (6), (7) and (8) which lines are accessible? which subderivatrions (if any) are accessible?

	accessible lines	accessible subderivations
line 6	(1), (4), (5)	2-3

E6.4. Suppose in a derivation with structure as in E6.3 we have obtained a formula $\mathcal{A}$ on line (3). (i) On what lines would we be allowed to conclude $\mathcal{A}$ by 3 R? Suppose there is a formula $\mathcal{B}$ on line (4). (ii) On what lines would be be allowed to conclude $\mathcal{B}$ by 4 R?

(i) There are no lines on which we could conclude $\mathcal{A}$ by 3 R.

E6.6. The following are not legitimate *ND* derivations. In each case, explain why.

a.

$$\begin{array}{lll} 1. & (A \wedge B) \wedge (C \to B) & \text{P} \\ 2. & A & 1\ \wedge\text{E} \end{array}$$

This does not apply the rule to the main operator. From (1) by $\wedge$E we can get $A \wedge B$ or $C \to B$. From the first A would follow by a *second* application of the rule.

E6.7. Provide derivations to show each of the following.

b. $A \wedge B, B \to C \vdash_{ND} C$

$$\begin{array}{lll} 1. & A \wedge B & \text{P} \\ 2. & B \to C & \text{P} \\ 3. & B & 1\ \wedge\text{E} \\ 4. & C & 2,3\ \to\text{E} \end{array}$$

e. $A \to (A \to B) \vdash_{ND} A \to B$

$$\begin{array}{lll} 1. & A \to (A \to B) & \text{P} \\ 2. & \quad A & \text{A } (g, \to\text{I}) \\ 3. & \quad A \to B & 1,2\ \to\text{E} \\ 4. & \quad B & 3,2\ \to\text{E} \\ 5. & A \to B & 2\text{-}4\ \to\text{I} \end{array}$$

h. $A \to B, B \to C \vdash_{ND} (A \wedge K) \to C$

$$\begin{array}{lll} 1. & A \to B & \text{P} \\ 2. & B \to C & \text{P} \\ 3. & \quad A \wedge K & \text{A } (g, \to\text{I}) \\ 4. & \quad A & 3\ \wedge\text{E} \\ 5. & \quad B & 1,4\ \to\text{E} \\ 6. & \quad C & 2,5\ \to\text{E} \\ 7. & (A \wedge K) \to C & 3\text{-}6\ \to\text{I} \end{array}$$

l. $A \to B \vdash_{ND} (C \to A) \to (C \to B)$

$$\begin{array}{lll} 1. & A \to B & \text{P} \\ 2. & \quad C \to A & \text{A } (g, \to\text{I}) \\ 3. & \quad\quad C & \text{A } (g, \to\text{I}) \\ 4. & \quad\quad A & 2,3\ \to\text{E} \\ 5. & \quad\quad B & 1,4\ \to\text{E} \\ 6. & \quad C \to B & 3\text{-}5\ \to\text{I} \\ 7. & (C \to A) \to (C \to B) & 2\text{-}6\ \to\text{I} \end{array}$$

Exercise 6.7.l

E6.9. The following are not legitimate *ND* derivations. In each case, explain why.

c.

1.	$\vert\ \underline{W}$	P
2.	$\vert\ \vert\ \underline{R}$	A (c, $\sim$I)
3.	$\vert\ \vert\ \vert\ \underline{\sim W}$	A (c, $\sim$I)
4.	$\vert\ \vert\ \vert\ \bot$	1,3 $\bot$I
5.	$\vert\ \sim R$	2-4 $\sim$I

There is no contradiction against the scope line for assumption R. So we are not justified in exiting the subderivation that begins on (2). The contradiction *does* justify exiting the subderivation that begins on (3) with the conclusion W by 3-4 $\sim$E. But this would still be under the scope of assumption R, and does not get us anywhere, as we already had W at line (1)!

E6.10. Produce derivations to show each of the following.

c. $\sim A \rightarrow B, \sim B \vdash_{ND} A$

1.	$\vert\ \sim A \rightarrow B$	P
2.	$\vert\ \underline{\sim B}$	P
3.	$\vert\ \vert\ \underline{\sim A}$	A (c, $\sim$E)
4.	$\vert\ \vert\ B$	1,3 $\rightarrow$E
5.	$\vert\ \vert\ \bot$	4,2 $\bot$I
6.	$\vert\ A$	3-5 $\sim$E

g. $A \vee (A \wedge B) \vdash_{ND} A$

1.	$\vert\ \underline{A \vee (A \wedge B)}$	P
2.	$\vert\ \vert\ \underline{A}$	A (g, 1$\vee$E)
3.	$\vert\ \vert\ A$	2 R
4.	$\vert\ \vert\ \underline{A \wedge B}$	A (g, 1$\vee$E)
5.	$\vert\ \vert\ A$	4 $\wedge$E
6.	$\vert\ A$	1,2-3,4-5 $\vee$E

l. $A \to \sim B \vdash_{ND} B \to \sim A$

1.	$A \to \sim B$	P
2.	B	A (g, $\to$I)
3.	A	A (c, $\sim$I)
4.	$\sim B$	1,3 $\to$E
5.	$\bot$	2,4 $\bot$I
6.	$\sim A$	3-5 $\sim$I
7.	$B \to \sim A$	2-6 $\to$I

E6.12. Each of the following are not legitimate *ND* derivations. In each case, explain why.

c.

1.	$A \leftrightarrow B$	P
2.	A	1 $\leftrightarrow$E

$\leftrightarrow$E takes as inputs a biconditional *and* one side or the other. We cannot get A from (1) unless we already have B.

E6.13. Produce derivations to show each of the following.

a. $(A \wedge B) \leftrightarrow A \vdash_{ND} A \to B$

1.	$(A \wedge B) \leftrightarrow A$	P
2.	A	A (g, $\to$I)
3.	$A \wedge B$	1,2 $\leftrightarrow$E
4.	B	3 $\wedge$E
5.	$A \to B$	2-4 $\to$I

e. $A \leftrightarrow (B \wedge C), B \vdash_{ND} A \leftrightarrow C$

1.	$A \leftrightarrow (B \wedge C)$	P
2.	B	P
3.	A	A (g, $\leftrightarrow$I)
4.	$B \wedge C$	1,3 $\leftrightarrow$E
5.	C	4 $\wedge$E
6.	C	A (g, $\leftrightarrow$I)
7.	$B \wedge C$	2,6 $\wedge$I
8.	A	1,7 $\leftrightarrow$E
9.	$A \leftrightarrow C$	3-5,6-8 $\leftrightarrow$I

Exercise 6.13.e

k. $\vdash_{ND} \sim\sim A \leftrightarrow A$

1.	$\sim\sim A$	A (g, $\leftrightarrow$I)
2.	$\sim A$	A (c, $\sim$E)
3.	$\sim\sim A$	1 R
4.	$\perp$	2,3 $\perp$I
5.	A	2-4 $\sim$E
6.	A	A (g $\leftrightarrow$I)
7.	$\sim A$	A (g, $\sim$I)
8.	A	6 R
9.	$\perp$	8,7 $\perp$I
10.	$\sim\sim A$	7-9 $\sim$I
11.	$\sim\sim A \leftrightarrow A$	1-5,6-10 $\leftrightarrow$I

E6.14. For each of the following, (i) which primary strategy applies? and (ii) what is the next step? If the strategy calls for a new subgoal, show the subgoal; if it calls for a subderivation, set up the subderivation. In each case, *explain* your response.

c.

1.	$\sim A \leftrightarrow B$	P
	$B \leftrightarrow \sim A$	

(i) There is no contradiction in accessible lines so SG1 does not apply. There is no disjunction in accessible lines so SG2 does not apply. The goal does not appear in the premises so SG3 does not apply. (ii) Given this, we apply SG4 and go for the goal by $\leftrightarrow$I. For this goal $\leftrightarrow$I requires a pair of subderivations which set up as follows.

1.	$\sim A \leftrightarrow B$	P
2.	B	A (g $\leftrightarrow$I)
	$\sim A$	
	$\sim A$	A (g $\leftrightarrow$I)
	B	
	$B \leftrightarrow \sim A$	_,_ $\leftrightarrow$I

E6.15. Produce derivations to show each of the following. No worked out answers are provided. However, if you get stuck, you will find strategy hints in the back.

a. $A \leftrightarrow (A \rightarrow B) \vdash_{ND} A \rightarrow B$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\rightarrow$I in application of SG4.

b. $(A \vee B) \rightarrow (B \leftrightarrow D), B \vdash_{ND} B \wedge D$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So plan to get the primary goal by $\wedge$I in application of SG4. Then it is a matter of SG3 to get the parts.

c. $\sim(A \wedge C), \sim(A \wedge C) \leftrightarrow B \vdash_{ND} A \vee B$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So plan to get the primary goal by (one form of) $\vee$I in application of SG4.

d. $A \wedge (C \wedge \sim B), (A \vee D) \rightarrow \sim E \vdash_{ND} \sim E$

Hint: There is no contradiction or disjunction; but the goal exists in the premises. So proceed by application of SG3.

e. $A \rightarrow B, B \rightarrow C \vdash_{ND} A \rightarrow C$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\rightarrow$I in application of SG4.

f. $(A \wedge B) \rightarrow (C \wedge D) \vdash_{ND} [(A \wedge B) \rightarrow C] \wedge [(A \wedge B) \rightarrow D]$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\wedge$I in application of SG4. Then apply SG4 and $\rightarrow$I again for your new subgoals.

g. $A \rightarrow (B \rightarrow C), (A \wedge D) \rightarrow E, C \rightarrow D \vdash_{ND} (A \wedge B) \rightarrow E$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\rightarrow$I in application of SG4. Then it is a matter of SG3.

h. $(A \rightarrow B) \wedge (B \rightarrow C), [(D \vee E) \vee H] \rightarrow A, \sim(D \vee E) \wedge H \vdash_{ND} C$

Hint: There is no contradiction or disjunction; but the goal is in the premises. So proceed by application of SG3.

Exercise 6.15.h

i. $A \to (B \wedge C), \sim C \vdash_{ND} \sim(A \wedge D)$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\sim$I in application of SG4.

j. $A \to (B \to C), D \to B \vdash_{ND} A \to (D \to C)$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\to$I in application of SG4. Similar reasoning applies to the secondary goal.

k. $A \to (B \to C) \vdash_{ND} \sim C \to \sim(A \wedge B)$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\to$I in application of SG4. You can also apply SG4 to the secondary goal.

l. $(A \wedge \sim B) \to \sim A \vdash_{ND} A \to B$

Hint: There is no simple contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\to$I in application of SG4. This time the secondary goal has no operator, and so falls all the way through to SG5.

m. $\sim B \leftrightarrow A, C \to B, A \wedge C \vdash_{ND} \sim K$

Hint: There is no contradiction or disjunction; and the goal is not in the premises. So set up to get the primary goal by $\sim$I in application of SG4. This works because the premises are themselves inconsistent.

n. $\sim A \vdash_{ND} A \to B$

Hint: After you set up for the main goal, look for an application of SG1.

o. $\sim A \leftrightarrow \sim B \vdash_{ND} A \leftrightarrow B$

Hint: After you set up for the main goal, look for applications of SG5.

p. $(A \vee B) \vee C, B \leftrightarrow C \vdash_{ND} C \vee A$

Hint: This is not hard, if you recoginize each of the places where SG2 applies.

q. $\vdash_{ND} A \to (A \vee B)$

Hint: Do not panic. Without premises, there is definately no contradiction or disjunction; and the goal is not in accessible lines! So set up to get the primary goal by $\to$I in application of SG4.

Exercise 6.15.q

r. $\vdash_{ND} A \rightarrow (B \rightarrow A)$

Hint: Apply SG4 to get the goal, and again for the subgoal.

s. $\vdash_{ND} (A \leftrightarrow B) \rightarrow (A \rightarrow B)$

Hint: This requires multiple applications of SG4.

t. $\vdash_{ND} (A \wedge \sim A) \rightarrow (B \wedge \sim B)$

Hint: Once you set up for the main goal, look for an application of SG1.

u. $\vdash_{ND} (A \rightarrow B) \rightarrow [(C \rightarrow A) \rightarrow (C \rightarrow B)]$

Hint: This requires multiple applications of SG4.

v. $\vdash_{ND} [(A \rightarrow B) \wedge \sim B] \rightarrow \sim A$

Hint: Apply SG4 to get the main goal, and again to get the subgoal.

w. $\vdash_{ND} A \rightarrow [B \rightarrow (A \rightarrow B)]$

Hint: This requires multiple applications of SG4.

x. $\vdash_{ND} \sim A \rightarrow [(B \wedge A) \rightarrow C]$

Hint: After a couple applications of SG4, you will have occaision to make use of SG1 — or equivalently, SG5.

y. $\vdash_{ND} (A \rightarrow B) \rightarrow [\sim B \rightarrow \sim(A \wedge D)]$

Hint: This requires multiple applications of SG4.

E6.16. Produce derivations to demonstrate each of T6.1 - T6.18.

T6.3. $\vdash_{ND} (\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$

1.	$\sim\mathcal{Q} \rightarrow \sim\mathcal{P}$	A (g, $\rightarrow$I)
2.	$\sim\mathcal{Q} \rightarrow \mathcal{P}$	A (g, $\rightarrow$I)
3.	$\sim\mathcal{Q}$	A (c, $\sim$E)
4.	$\mathcal{P}$	2,3 $\rightarrow$E
5.	$\sim\mathcal{P}$	1,3 $\rightarrow$E
6.	$\perp$	4,5 $\perp$I
7.	$\mathcal{Q}$	3-6 $\sim$E
8.	$(\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q}$	2-7 $\rightarrow$I
9.	$(\sim\mathcal{Q} \rightarrow \sim\mathcal{P}) \rightarrow ((\sim\mathcal{Q} \rightarrow \mathcal{P}) \rightarrow \mathcal{Q})$	1-8 $\rightarrow$I

T6.11. $\vdash_{ND} (\mathscr{A} \vee \mathscr{B}) \leftrightarrow (\mathscr{B} \vee \mathscr{A})$

1.	$\mathscr{A} \vee \mathscr{B}$	A (g, $\leftrightarrow$I)
2.	$\mathscr{A}$	A (g, 1$\vee$E)
3.	$\mathscr{B} \vee \mathscr{A}$	2 $\vee$I
4.	$\mathscr{B}$	A (g, 1$\vee$E)
5.	$\mathscr{B} \vee \mathscr{A}$	4 $\vee$I
6.	$\mathscr{B} \vee \mathscr{A}$	1,2-3,4-5 $\vee$E
7.	$\mathscr{B} \vee \mathscr{A}$	A (g, $\leftrightarrow$I)
8.	$\mathscr{B}$	A (g, 7$\vee$E)
9.	$\mathscr{A} \vee \mathscr{B}$	8 $\vee$I
10.	$\mathscr{A}$	A (g, 7$\vee$E)
11.	$\mathscr{A} \vee \mathscr{B}$	10 $\vee$I
12.	$\mathscr{A} \vee \mathscr{B}$	7,8-9,10-11 $\vee$E
13.	$(\mathscr{A} \vee \mathscr{B}) \leftrightarrow (\mathscr{B} \vee \mathscr{A})$	1-6,7-12 $\leftrightarrow$I

E6.17. Each of the following begins with a simple application of $\sim$I or $\sim$E for SG4 or SG5. Complete the derivations, and *explain* your use of secondary strategy.

a.

1.	$A \wedge B$	P
2.	$\sim(A \wedge C)$	P
3.	C	A (c, $\sim$I)
	$\bot$	
	$\sim C$	

1.	$A \wedge B$	P
2.	$\sim(A \wedge C)$	P
3.	C	A (c, $\sim$I)
4.	A	1 $\wedge$E
5.	$A \wedge C$	4,3 $\wedge$I
6.	$\bot$	5,2 $\bot$I
7.	$\sim C$	3-6 $\sim$I

There is no contradiction by atomics and negated atomics. And there is no disjunction in the scope of the assumption for $\sim$I. So we fall through to SC3. For this set the opposite of (2) as goal, and use primary strategies for it. The derivation of $A \wedge C$ is easy.

E6.18. Produce derivations to show each of the following. No worked out answers are provided. However, if you get stuck, you will find strategy hints in the back.

a. $A \to \sim(B \wedge C)$, $B \to C \vdash_{ND} A \to \sim B$

Apply primary strategies for $\to$I and $\sim$I. Then there will be occasion for a simple application of SC3.

b. $\vdash_{ND} \sim(A \to A) \to A$

Apply primary strategies for $\to$I and $\sim$E. Then there will be occasion for a simple application of SC3.

c. $A \vee B \vdash_{ND} \sim(\sim A \wedge \sim B)$

This requires no more than SC1, if you follow the primary strategies properly. From the start, apply sg2 to go for the whole goal $\sim(\sim A \wedge \sim B)$ by $\vee$E.

d. $\sim(A \wedge B)$, $\sim(A \wedge \sim B) \vdash_{ND} \sim A$

You will go for the main goal by $\sim$I in an instance of SG4. Then it is easiest to see this as a case where you use the premises for separate instances of SC3. It is, however, also possible to see the derivation along the lines of SC4.

e. $\vdash_{ND} A \vee \sim A$

For your primary strategy, fall all the way through to SG5. Then you will be able to see the derivation either along the lines of SC3 or 4, building up to the opposite of $\sim(A \vee \sim A)$ twice.

f. $\vdash_{ND} A \vee (A \to B)$

Your primary strategy falls through to SG5. Then $\sim A$ is sufficient to prove $A \to B$, and this turns into a pure version of the pattern (AQ) for formulas with main operator $\vee$.

g. $A \vee \sim B$, $\sim A \vee \sim B \vdash_{ND} \sim B$

For this you will want to apply SG2 to one of the premises (it does not matter which) for the goal. This gives you a pair of subderivations. One is easy. In the other, SG2 applies again!

h. $A \leftrightarrow (\sim B \vee C)$, $B \to C \vdash_{ND} A$

The goal is in the premises, so your primary strategy is SG3. The real challenge is getting $\sim B \vee C$. For this you will fall through to SG5, and assume its negation. Then the derivation can be conceived either along the lines of SC3 or SC4, and on the standard pattern for disjunctions.

i. $A \leftrightarrow B \vdash_{ND} (C \leftrightarrow A) \leftrightarrow (C \leftrightarrow B)$

Applying SG4, set up for the primary goal by $\leftrightarrow$I. You will then need $\leftrightarrow$I for the subgoals as well.

j. $A \leftrightarrow \sim(B \leftrightarrow \sim C), \sim(A \vee B) \vdash_{ND} C$

Fall through to SG5 for the primary goal. Then you can think of the derivation along the lines of SC3 or SC4. The derivation of $A \vee B$ works on the standard pattern, insofar as with the assumption $\sim C$, $\sim A$ gets you B.

k. $[C \vee (A \vee B)] \wedge (C \rightarrow E), A \rightarrow D, D \rightarrow \sim A \vdash_{ND} C \vee B$

Though officially there is no formula with main operator $\vee$, a minor reshuffle exposes $C \vee (A \vee B)$ on an accessible line. Then the derivation is naturally driven by applications of SG2.

l. $\sim(A \rightarrow B), \sim(B \rightarrow C) \vdash_{ND} \sim D$

Go for the main goal by $\sim$I in applicaiton of SG4. Then it is most natural to see the derivation as involving two separate applications of SC3. It is also possible to set the derivation up along the lines of SC4, though this leads to a rather different result.

m. $C \rightarrow \sim A, \sim(B \wedge C) \vdash_{ND} (A \vee B) \rightarrow \sim C$

Go for the primary goal by $\rightarrow$I in application of SG4. Then you will need to apply SG2 to reach the subgoal.

n. $\sim(A \leftrightarrow B) \vdash_{ND} \sim A \leftrightarrow B$

Go for the primary goal by $\leftrightarrow$I in application of SG4. You can go for one subgoal by $\sim$E, the other by $\sim$I. Then fall through to SC3 for the conradictions, where this will involve you in further instances of $\leftrightarrow$I. The derivation is long, but should be straightforward if you follow the strategies.

o. $A \leftrightarrow B, B \leftrightarrow \sim C \vdash_{ND} \sim(A \leftrightarrow C)$

Go for the primary goal by $\sim$I in application of SG4. Then the contradiction comes by application of SC4.

p. $A \vee B, \sim B \vee C, \sim C \vdash_{ND} A$

This will set up as a couple instances of $\vee$E. If you begin with $A \vee B$, one subderivation is easy. In the second, be on the lookout for a couple instances of SG1.

q. $(\sim A \vee C) \vee D,\ D \rightarrow \sim B \vdash_{ND} (A \wedge B) \rightarrow C$

Officially, the primary strategy should be $\vee$E in application of SG2. However, in this case it will not hurth to begin with $\rightarrow$I, and set up $\vee$E inside the subderivation for that.

r. $A \vee D,\ \sim D \leftrightarrow (E \vee C),\ (C \wedge B) \vee [C \wedge (F \rightarrow C)] \vdash_{ND} A$

The two disjunctions require applications of SG2. In fact, there are ways to simplify this from the mechanical version entirely driven by the strategy.

s. $(A \vee B) \vee (C \wedge D),\ (A \leftrightarrow E) \wedge (B \rightarrow F),\ G \leftrightarrow \sim(E \vee F),\ C \rightarrow B \vdash_{ND} \sim G$

This derivation is driven by $\vee$E in application of SG2 and then SC3. Again, there are ways to make the derivation relatively more elegant.

t. $(A \vee B) \wedge \sim C,\ \sim C \rightarrow (D \wedge \sim A),\ B \rightarrow (A \vee E) \vdash_{ND} E \vee F$

Since there is no F in the premises, it makes sense to think the conclusion is true because E is true. So it is safe to set up to get the conclusion from E by $\vee$I. After some simplification, the overall strategy is revealed to be $\vee$E based on $A \vee B$, in application of SG2. One subderivation has another formula with main operator $\vee$, and so another instance of $\vee$E.

E6.19. Produce derivations to demonstrate each of T6.19 - T6.26.

T6.19. $\vdash_{ND} \sim(\mathcal{A} \wedge \mathcal{B}) \leftrightarrow (\sim\mathcal{A} \vee \sim\mathcal{B})$

1.	$\sim(\mathcal{A} \wedge \mathcal{B})$	A (g, $\leftrightarrow$I)
2.	$\sim(\sim\mathcal{A} \vee \sim\mathcal{B})$	A (c, $\sim$E)
3.	$\sim\mathcal{A}$	A (c, $\sim$E)
4.	$\sim\mathcal{A} \vee \sim\mathcal{B}$	3 $\vee$I
5.	$\perp$	4,2 $\perp$I
6.	$\mathcal{A}$	3-5 $\sim$E
7.	$\sim\mathcal{B}$	A (c, $\sim$E)
8.	$\sim\mathcal{A} \vee \sim\mathcal{B}$	7 $\vee$I
9.	$\perp$	8,2 $\perp$I
10.	$\mathcal{B}$	7-9 $\sim$E
11.	$\mathcal{A} \wedge \mathcal{B}$	6,10 $\wedge$I
12.	$\perp$	11,1 $\perp$I
13.	$\sim\mathcal{A} \vee \sim\mathcal{B}$	2-12 $\sim$E
14.	$\sim\mathcal{A} \vee \sim\mathcal{B}$	A (g, $\leftrightarrow$I)
15.	$\sim\mathcal{A}$	A (g, 14$\vee$E)
16.	$\mathcal{A} \wedge \mathcal{B}$	A (c, $\sim$I)
17.	$\mathcal{A}$	16 $\wedge$E
18.	$\perp$	17,15 $\perp$I
19.	$\sim(\mathcal{A} \wedge \mathcal{B})$	16-18 $\sim$I
20.	$\sim\mathcal{B}$	A (g, 14$\vee$E)
21.	$\mathcal{A} \wedge \mathcal{B}$	A (c, $\sim$I)
22.	$\mathcal{B}$	21 $\wedge$E
23.	$\perp$	22,20 $\perp$I
24.	$\sim(\mathcal{A} \wedge \mathcal{B})$	21-23 $\sim$I
25.	$\sim(\mathcal{A} \wedge \mathcal{B})$	14,15-19,20-24 $\vee$E
26.	$\sim(\mathcal{A} \wedge \mathcal{B}) \leftrightarrow (\sim\mathcal{A} \vee \sim\mathcal{B})$	1-13,14-25 $\leftrightarrow$I

Chapter Seven

E7.1. Suppose $\mathsf{I}[A] = \mathsf{T}$, $\mathsf{I}[B] \neq \mathsf{T}$ and $\mathsf{I}[C] = \mathsf{T}$. For each of the following, produce a formalized derivation, and then non-formalized reasoning to demonstrate either that it is or is not true on I.

b. $\mathsf{I}[\sim B \rightarrow \sim C] \neq \mathsf{T}$

1. $\mathsf{I}[B] \neq \mathsf{T}$	prem	
2. $\mathsf{I}[\sim B] = \mathsf{T}$	1 ST($\sim$)	
3. $\mathsf{I}[C] = \mathsf{T}$	prem	
4. $\mathsf{I}[\sim C] \neq \mathsf{T}$	3 ST($\sim$)	
5. $\mathsf{I}[\sim B] = \mathsf{T} \vartriangle \mathsf{I}[\sim C] \neq \mathsf{T}$	2,4 cnj	
6. $\mathsf{I}[\sim B \rightarrow \sim C] \neq \mathsf{T}$	5 ST($\rightarrow$)	

It is given that $\mathsf{I}[B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim B] = \mathsf{T}$. But it is given that $\mathsf{I}[C] = \mathsf{T}$; so by ST($\sim$), $\mathsf{I}[\sim C] \neq \mathsf{T}$. So $\mathsf{I}[\sim B] = \mathsf{T}$ and $\mathsf{I}[\sim C] \neq \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{I}[\sim B \rightarrow \sim C] \neq \mathsf{T}$.

E7.2. Produce a formalized derivation, and then informal reasoning to demonstrate each of the following.

a. $A \rightarrow B, \sim A \not\models_s \sim B$

Set $\mathsf{J}[A] \neq \mathsf{T}$, $\mathsf{J}[B] = \mathsf{T}$

1. $\mathsf{J}[A] \neq \mathsf{T}$	ins (J particular)
2. $\mathsf{J}[\sim A] = \mathsf{T}$	1 ST($\sim$)
3. $\mathsf{J}[A] \neq \mathsf{T} \triangledown \mathsf{J}[B] = \mathsf{T}$	1 dsj
4. $\mathsf{J}[A \rightarrow B] = \mathsf{T}$	3 ST($\rightarrow$)
5. $\mathsf{J}[B] = \mathsf{T}$	ins
6. $\mathsf{J}[\sim B] \neq \mathsf{T}$	5 ST($\sim$)
7. $\mathsf{J}[A \rightarrow B] = \mathsf{T} \vartriangle \mathsf{J}[\sim A] = \mathsf{T} \vartriangle \mathsf{J}[\sim B] \neq \mathsf{T}$	4,2,6 cnj
8. $S\mathsf{I}(\mathsf{I}[A \rightarrow B] = \mathsf{T} \vartriangle \mathsf{I}[\sim A] = \mathsf{T} \vartriangle \mathsf{I}[\sim B] \neq \mathsf{T})$	7 exs
9. $A \rightarrow B, \sim A \not\models_s \sim B$	8 SV

$\mathsf{J}[A] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[\sim A] = \mathsf{T}$. But since $\mathsf{J}[A] \neq \mathsf{T}$, $\mathsf{J}[A] \neq \mathsf{T}$ or $\mathsf{J}[B] = \mathsf{T}$; so by ST($\rightarrow$), $\mathsf{J}[A \rightarrow B] = \mathsf{T}$. And $\mathsf{J}[B] = \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[\sim B] \neq \mathsf{T}$. So $\mathsf{J}[A \rightarrow B] = \mathsf{T}$, and $\mathsf{J}[\sim A] = \mathsf{T}$, but $\mathsf{J}[\sim B] \neq \mathsf{T}$; so there is an interpretation I such that $\mathsf{I}[A \rightarrow B] = \mathsf{T}$, and $\mathsf{I}[\sim A] = \mathsf{T}$, but $\mathsf{I}[\sim B] \neq \mathsf{T}$; so by SV, $A \rightarrow B, \sim A \not\models_s \sim B$.

b. $A \rightarrow B, \sim B \models_s \sim A$

1. $\quad A \rightarrow B, \sim B \not\models_s \sim A$	assp
2. $\quad S\mathsf{I}(\mathsf{I}[A \rightarrow B] = \mathsf{T} \vartriangle \mathsf{I}[\sim B] = \mathsf{T} \vartriangle \mathsf{I}[\sim A] \neq \mathsf{T})$	1 SV
3. $\quad \mathsf{J}[A \rightarrow B] = \mathsf{T} \vartriangle \mathsf{J}[\sim B] = \mathsf{T} \vartriangle \mathsf{J}[\sim A] \neq \mathsf{T}$	2 exs (J particular)
4. $\quad \mathsf{J}[\sim B] = \mathsf{T}$	3 cnj
5. $\quad \mathsf{J}[B] \neq \mathsf{T}$	4 ST($\sim$)
6. $\quad \mathsf{J}[A \rightarrow B] = \mathsf{T}$	3 cnj
7. $\quad \mathsf{J}[A] \neq \mathsf{T} \triangledown \mathsf{J}[B] = \mathsf{T}$	6 ST($\rightarrow$)
8. $\quad \mathsf{J}[A] \neq \mathsf{T}$	7,5 dsj
9. $\quad \mathsf{J}[\sim A] \neq \mathsf{T}$	3 cnj
10. $\quad \mathsf{J}[A] = \mathsf{T}$	9 ST($\sim$)
11. $A \rightarrow B, \sim B \models_s \sim A$	1-9 neg

Suppose $A \rightarrow B, \sim B \not\models_s \sim A$; then by SV there is an I such that $\mathsf{I}[A \rightarrow B] = \mathsf{T}$ and $\mathsf{I}[\sim B] = \mathsf{T}$ and $\mathsf{I}[\sim A] \neq \mathsf{T}$. Let J be a particular interpretation of this sort;

then $\mathsf{J}[A \to B] = \mathsf{T}$ and $\mathsf{J}[\sim B] = \mathsf{T}$ and $\mathsf{J}[\sim A] \neq \mathsf{T}$. Since $\mathsf{J}[\sim B] = \mathsf{T}$, by ST($\sim$), $\mathsf{J}[B] \neq \mathsf{T}$. And since $\mathsf{J}[A \to B] = \mathsf{T}$, either $\mathsf{J}[A] \neq \mathsf{T}$ or $\mathsf{J}[B] = \mathsf{T}$; so $\mathsf{J}[A] \neq \mathsf{T}$. But since $\mathsf{J}[\sim A] \neq \mathsf{T}$, by ST($\sim$), $\mathsf{J}[A] = \mathsf{T}$. This is impossible; reject the assumption: $A \to B, \sim B \vDash_s \sim A$.

E7.4. Complete the demonstration of derived clauses ST′ by completing the demonstration for dst in the other direction (and providing demonstrations for other clauses).

1.	$[(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})] \vartriangle \neg[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})]$	assp
2.	$(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	1 cnj
3.	$\neg[(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})]$	1 cnj
4.	$\neg(\neg\mathfrak{A} \triangledown \mathfrak{B}) \triangledown \neg(\neg\mathfrak{B} \triangledown \mathfrak{A})$	3 dem
5.	$\quad \neg\mathfrak{A} \triangledown \mathfrak{B}$	assp
6.	$\quad \neg(\neg\mathfrak{B} \triangledown \mathfrak{A})$	4,5 dsj
7.	$\quad \mathfrak{B} \vartriangle \neg\mathfrak{A}$	6 dem
8.	$\quad \mathfrak{B}$	7 cnj
9.	$\quad \mathfrak{A} \triangledown \mathfrak{B}$	8 dsj
10.	$\quad \neg(\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	9 dem
11.	$\quad \mathfrak{A} \vartriangle \mathfrak{B}$	2,10 dsj
12.	$\quad \mathfrak{A}$	11 cnj
13.	$\quad \neg\mathfrak{A}$	7 cnj
14.	$\neg(\neg\mathfrak{A} \triangledown \mathfrak{B})$	5-13 neg
15.	$\mathfrak{A} \vartriangle \neg\mathfrak{B}$	14 dem
16.	$\mathfrak{A}$	15 cnj
17.	$\mathfrak{A} \triangledown \mathfrak{B}$	16 dsj
18.	$\neg(\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})$	17 dem
19.	$\mathfrak{A} \vartriangle \mathfrak{B}$	2,18 dsj
20.	$\mathfrak{B}$	19 cnj
21.	$\neg\mathfrak{B}$	15 cnj
22.	$[(\mathfrak{A} \vartriangle \mathfrak{B}) \triangledown (\neg\mathfrak{A} \vartriangle \neg\mathfrak{B})] \Rightarrow [(\neg\mathfrak{A} \triangledown \mathfrak{B}) \vartriangle (\neg\mathfrak{B} \triangledown \mathfrak{A})]$	1-22 cnd

E7.5. Using ST($\mid$) as on p. 333, produce non-formalized reasonings to show each of the following.

b. $\mathsf{I}[\mathscr{P} \mid (\mathscr{Q} \mid \mathscr{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P} \to \mathscr{Q}] = \mathsf{T}$

By ST($\mid$), $\mathsf{I}[\mathscr{P} \mid (\mathscr{Q} \mid \mathscr{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$ or $\mathsf{I}[\mathscr{Q} \mid \mathscr{Q}] \neq \mathsf{T}$; by ST($\mid$), iff $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$ or ($\mathsf{I}[Q] = \mathsf{T}$ and $\mathsf{I}[Q] = \mathsf{T}$); iff iff $\mathsf{I}[\mathscr{P}] \neq \mathsf{T}$ or $\mathsf{I}[Q] = \mathsf{T}$; by ST($\to$), iff $\mathsf{I}[\mathscr{P} \to \mathscr{Q}] = \mathsf{T}$. So $\mathsf{I}[\mathscr{P} \mid (\mathscr{Q} \mid \mathscr{Q})] = \mathsf{T}$ iff $\mathsf{I}[\mathscr{P} \to \mathscr{Q}] = \mathsf{T}$.

E7.6. Produce non-formalized reasoning to demonstrate each of the following.

b. $\sim(A \leftrightarrow B), \sim A, \sim B \vDash_s C \wedge \sim C$

Suppose $\sim(A \leftrightarrow B)$, $\sim A$, $\sim B \nvDash_s C \wedge \sim C$; then by SV there is some I such that $\mathsf{I}[\sim(A \leftrightarrow B)] = \mathsf{T}$, and $\mathsf{I}[\sim A] = \mathsf{T}$, and $\mathsf{I}[\sim B] = \mathsf{T}$, but $\mathsf{I}[C \wedge \sim C] \neq \mathsf{T}$. Let J be a particular interpretation of this sort; then $\mathsf{J}[\sim(A \leftrightarrow B)] = \mathsf{T}$, and $\mathsf{J}[\sim A] = \mathsf{T}$, and $\mathsf{J}[\sim B] = \mathsf{T}$, but $\mathsf{J}[C \wedge \sim C] \neq \mathsf{T}$. From the first, by ST($\sim$), $\mathsf{J}[A \leftrightarrow B] \neq \mathsf{T}$; so by ST$'$($\leftrightarrow$), ($\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$) or ($\mathsf{J}[A] \neq \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$). But since $\mathsf{J}[\sim A] = \mathsf{T}$, by ST($\sim$), $\mathsf{J}[A] \neq \mathsf{T}$; so $\mathsf{J}[A] \neq \mathsf{T}$ or $\mathsf{J}[B] = \mathsf{T}$; so it is not the case that $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$; so $\mathsf{J}[A] \neq \mathsf{T}$ and $\mathsf{J}[B] = \mathsf{T}$; so $\mathsf{J}[B] = \mathsf{T}$. But $\mathsf{J}[\sim B] = \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[B] \neq \mathsf{T}$. This is impossible; reject the assumption: $\sim(A \leftrightarrow B)$, $\sim A$, $\sim B \vDash_s C \wedge \sim C$.

c. $\sim(\sim A \wedge \sim B) \nvDash_s A \wedge B$

Set $\mathsf{J}[A] = \mathsf{T}$ and $\mathsf{J}[B] \neq \mathsf{T}$.

$\mathsf{J}[A] = \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[\sim A] \neq \mathsf{T}$; so $\mathsf{J}[\sim A] \neq \mathsf{T}$ or $\mathsf{J}[\sim B] \neq \mathsf{T}$; so by ST$'$($\wedge$), $\mathsf{J}[\sim A \wedge \sim B] \neq \mathsf{T}$; so by ST($\sim$), $\mathsf{J}[\sim(\sim A \wedge \sim B)] = \mathsf{T}$. But it is given that $\mathsf{J}[B] \neq \mathsf{T}$; so $\mathsf{J}[A] \neq \mathsf{T}$ or $\mathsf{J}[B] \neq \mathsf{T}$; so by ST$'$($\wedge$), $\mathsf{J}[A \wedge B] \neq \mathsf{T}$. So $\mathsf{J}[\sim(\sim A \wedge \sim B)] = \mathsf{T}$ and $\mathsf{J}[A \wedge B] \neq \mathsf{T}$; so by SV, $\sim(\sim A \wedge \sim B) \nvDash_s A \wedge B$.

Bibliography

Benacerraf, P., and H. Putnam. *Philosophy of Mathematics: Selected Readings*. Cambridge: Cambridge University Press, 1983, 2nd edition.

Bergmann, M., J. Moor, and J. Nelson. *The Logic Book*. New York: McGraw-Hill, 2004, 4th edition.

Berto, Francesco. *There's Something About Gödel: The Complete Guide to the Incompleteness Theorem*. Oxford: Wiley-Blackwell, 2009.

Black, Robert. "Proving Church's Thesis." *Philosophia Mathematica* 8 (2000): 244–258.

Boolos, G., J. Burgess, and R. Jeffrey. *Computability and Logic*. Cambridge: Cambridge University Press, 2002, 4th edition.

Boolos, George. *The Logic of Provability*. Cambridge: Cambridge University Press, 1993.

Cederblom, J, and D Paulsen. *Critical Reasoning*. Belmont: Wadsworth, 2005, 6th edition.

Church, Alonzo. "An Unsolvable Problem of Elementary Number Theory." *American Journal of Mathematics* 58 (1936): 345–363.

Dennett, Daniel, editor. *The Philosopher's Lexicon*. 1987. URL http://www.blackwellpublishing.com/lexicon/.

Drake, F., and Singh D. *Intermediate Set Theory*. Chichester, England: John Wiley & Sons, 1996.

Earman, J, and J. Norton. "Forever is a Day: Supertasks in Pitowsky and Malament-Hogarth Spacetimes." *Philosophy of Science* 60 (1993): 22–42.

Earman, John. *Bangs, Crunches, Whimpers, and Shrieks: Singularities and Acausalities in Relativistic Spacetimes*. New York: Oxford University Press, 1995.

Enderton, H. *Elements of Set Theory*. Boston: Academic Press, Inc., 1977.

Feferman, et al., editors. *Gödel's Collected Works: Vol I*. New York: Oxford University Press, 1986.

Fisher, A. *Formal Number Theory and Computability*. Oxford: Clarendon Press, 1982.

George, A, and D. Velleman. *Philosophies of Mathematics*. Blackwell Publishers, 2002.

Gödel, K. "On Formally Undecidable Propositions of *Principia Mathematica* and Related Systems I." In *Collected Works, Vol. I: Publications 1929-1936*, Oxford: Oxford University Press, 1986, 144–95.

Gödel, Kurt. "Die Vollständigkeit der Axiome des Logischen Funktionenkalküls." *Monatshefte für Mathematik und Physik* 37 (1930): 349–360.

von Heijenoort, editor. *From Frege to Gödel*. Cambridge: Harvard University Press, 1967.

Henkin, Leon. "The Completeness of the First-Order Functional Calculus." *Journal of Symbolic Logic* 14 (1949): 159–166.

———. "A Problem Concerning Provability." *Journal of Symbolic Logic* 17 (1952): 160.

Hodges, W. *A Shorter Model Theory*. Cambridge: Cambridge University Press, 1997.

Hogarth, Mark. "Does General Relativity Allow an Observer To View an Eternity In a Finite Time?" *Foundations of Physics Letters* 173–181.

Kolmogorov, and Uspenskii. "On the Definition of an Algorithm." *American Mathematical Society Translations* 29 (1963): 217–245.

Kripke, Saul. *Wittgenstein on Rules and Private Language: An Elementary Exposition*. Cambridge, Mass.: Harvard University Press, 1982.

Manzano, María. *Extensions of First Order Logic*. Cambridge: Cambridge University Press, 1996.

———. *Model Theory*. Oxford: Clarendon Press, 1999.

Mendelson, Elliott. *Introduction to Mathematical Logic*. New York: Chapman and Hall, 1997, 4th edition.

Pietroski, Paul. “Logical Form.” In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, 2009. Fall, 2009 edition. URL `http://plato.stanford.edu/archives/fall2009/entries/logical-form/`.

Plantinga, Alvin. *God, Freedom, and Evil*. Grand Rapids: Eerdmans, 1977.

Pohlers, W. *Proof Theory*. Berlin: Springer-Verlag, 1989.

Priest, Graham. *Non-Classical Logics*. Cambridge: Cambridge University Press, 2001.

Putnam, Hilary. *Reason, Truth and History*. Cambridge: Cambridge University Pres, 1981.

Raatikainen, Panu. “Gödel’s Incompleteness Theorems.” In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, 2015. Spring 2015 edition. URL `http://plato.stanford.edu/archives/spr2015/entries/goedel-incompleteness/`.

Robinson, R. “An Essentially Undecidable Axiom System.” *Proceedings of the International Congress of Mathematics* 1 (1950): 729–730.

Rosser, Barkley. “Extensions of Some Theorems of Gödel and Church.” *Journal of Symbolic Logic* 1 (1936): 230–235.

Roy, Tony. “Natural Derivations for Priest, *An Introduction to Non-Classical Logic*.” *The Australasian Journal of Logic* 47–192. URL `http://ojs.victoria.ac.nz/ajl/article/view/1779`.

———. “Modality.” In *The Continuum Companion to Metaphysics*, London: Continuum Publishing Group, 2012, 46–66.

Russell, B. “On Denoting.” *Mind* 14.

Shapiro, S. *Foundations Without Foundationalism: A Case for Second Order Logic*. Oxford: Clarendon Press, 1991.

———. *Thinking About Mathematics: The Philosophy of Mathematics*. Oxford: Oxford University Press, 2000.

———. "Philosophy of Mathematics and Its Logic: Introduction." In *The Oxford Handbook of Philosophy of Mathematics and Logic*, edited by S. Shapiro, Oxford: Oxford University Press, 2005, 3–28.

Smith, Peter. "Squeezing Arguments." *Analysis* 71 (2011): 22–30.

———. *An Introduction to Gödel's Theorems*. Cambridge: Cambridge University Press, 2013a, second edition.

———. "Teach Yourself Logic: A Study Guide.", 2013b. URL `http://www.logicmatters.net/tyl/`.

Szabo, M., editor. *The Collected Papers of Gerhard Gentzen*. Amsterdam: North-Holland, 1969.

Takeuti, G. *Proof Theory*. Amsterdam: North-Holland, 1975.

Tourlakis, George. *Lectures in Logic and Set Theory, Volume I: Mathematical Logic*. Cambridge: Cambridge University Press, 2003.

Turing, Alan. "On Computable Numbers, With an Application to the *Entscheidungsproblem*." *Proceedings of the London Mathematical Society* 42 (1936): 230–265.

Wang, Hao. "The axiomatization of Arithmetic." *Journal of Symbolic Logic* 22 (1957): 145–158.

Index