Mathematical Logic

Richard Grandy · Michael Barkasi · Joshua Reagan

This book is set in Latin Modern (for text and math fonts, released under the GUST Font License) and Adobe Source Sans Pro (for headings, released by Adobe under the OFL).

Edition 1.0.0
January 12, 2026

Contents

Preface

What is mathematical logic?

Logic is the study of logical consequence and logical truth. Logical consequence is a relation between a set of sentences Δ (sometimes called 'premises') and a sentence Φ (sometimes called a 'conclusion') such that Φ 'follows from' Δ. For example, 'Fran is a Wallaby' is a logical consequence of 'If Fran was born last year, then Fran is a Wallaby' and 'Fran was born last year'. A sentence Φ is a logical truth if its truth can be known solely in virtue of logic. For example, 'If Fran was born last year, then Fran was born last year' seems true solely as a matter of logic.

These definitions are insufficiently precise for the needs of the scrupulous philosopher or mathematician. In *mathematical* logic we use mathematical methods to study logical consequence and logical truth. In this textbook we define abstract, formal languages which have as sentences strings of basic symbols. For sentences of these formal languages, we rigorously define (i) the relation of entailment and (ii) formal derivation systems. Both entailment and formal derivations provide a mathematical approximation of logical consequence. We also show how logical truth can be defined in terms of logical consequence. In rough terms, a logical truth can be understood as a sentence that follows from the empty set of sentences. The goal of this textbook is to give the reader an introduction to these mathematical definitions and methods.

What material is covered here?

The main subject of this text is classical first-order logic (or as we call it, quantificational logic), with sentential logic (sometimes called 'propositional' logic) introduced first. We focus mostly on a quantificational logic that doesn't include identity and function symbols. Our conjunction and disjunction operators are of arbitrary (but finite) arity because that more closely mirrors typical English use. Many-valued logic, modal logic, and the identity operator are briefly introduced in the last chapter. The main purpose there is to show how sentential and quantificational logic can be extended to treat other logical operators and to cover a wider range of applications.

The main goal of the text is to prove that for quantificational logic, whether one uses formal derivations or proofs of logical entailment, one always gets the same results. There are various ways to define the semantics for quantificational logic.[1] We use

[1] For example, game-theoretic semantics and substitutional quantifiers; see Dunn and Belnap 1968,

Benson Mates' modification of the standard Tarskian model-theoretic semantics.[2] The derivation system in this text is a natural deduction style system.[3] In the system used here only conditional elimination can discharge assumptions; this tends to make the other rules simpler. We use Fitch-style derivations,[4] but following Donald Kalish and Richard Montague[5] we discharge assumptions by drawing a box around each completed subproof. Rather than using the more commonly found Henkin-style proofs[6] to prove strong completeness, we prove it constructively. We believe this makes the connection between syntactic and semantic methods more intuitive. In Chapter 8 we provide an algorithm ("The Method") which, for any sentence Φ entailed by some set of sentences Δ, produces a derivation of Φ from Δ. The proof used here owes much to Willard Quine's completeness proof.[7]

How can I effectively use this book?

Reading mathematical texts is difficult, especially for those not accustomed to the style of prose typical of the genre. Following a mathematical argument requires significant cognitive effort and concentration, so prepare yourself. The most fruitful way to approach this text is with pencil and paper. As you read, work out the examples, write the proofs, and find counterexamples on your own. There's no reason to stick to the text—look for your own proofs and work out new examples. Effective use of the text requires active engagement.

We have tried our best to prepare a helpful and clear text. A large stock of examples is included and most intratext references include a page number. Many of the proofs in the text, especially the recursive proofs, leave steps for the reader to complete. These are always steps (or whole proofs) for which the reader will have seen something similar before. Many of the examples have been worked through in detail. Our hope is that the examples show clearly all the "moving parts"—that they show how the answers to problems follow directly from the definitions and theorems. You probably won't learn much logic by reading the text once or twice, so work through the examples and theorems.

Stevenson 1973, Hintikka 1996, Leblanc 2001, Westerståhl 2001, Jacquette 2002.

[2] Mates 1972

[3] This style of system was first developed by Gentzen in (1934), Hodges 2001b: 26. If you are curious, derivation systems roughly divide into four types: natural deduction, Hilbert-style axiomatic systems, Gentzen-style sequent calculi, and proof tableaux; see Hodges 2001a: 24 for an overview.

[4] Fitch 1952

[5] Kalish and Montague 1964

[6] Henkin 1949. For one of many modern treatments, see Bergmann et al. 2003: ch. 11.4.

[7] Quine 1982

Issues in the Philosophy of Logic

In mathematical logic, one begins with rigorous mathematical definitions of "sentence", truth, logical truth, entailment, and derivation and then proves theorems about them. A primary concern in logic is how to make sense of reasoning in natural languages and the notion of logical consequence. How the two—the rigorously defined mathematical concepts like entailment and "pretheoretic" concepts like logical consequence—relate is a natural question, and one that's received a great deal of attention by philosophers.[8]

Because the aim of this text is to introduce mathematical logic and not philosophical logic, we largely set these questions aside. In some places, usually for the sake of exposition or motivation, we make claims that raise issues squarely in the domain of philosophy of logic.[9] Given the aims and scope of this text, however, we aren't able to point out all the relevant philosophical concerns, let alone do much to motivate our potentially contentious claims.

History of Logic

Logic has a long history going back to Aristotle. Aristotle claims that he was the first to work out any systematic treatment of logical consequence (*Sophistical Refutations*, 34, 183b34–36; citation from Smith 1995: 27), and as Robin Smith notes, "we have no reason to dispute this" (1995: 27). Aristotle's work gave rise to a tradition of work in logic that extends through the Stoics, Byzantine commentators, early Islamic philosophy, and European Scholasticism. Modern mathematical logic has roots in this Aristotelian tradition as well (though it has roots elsewhere too, e.g. foundational work in mathematics). However, the Aristotelian tradition is beyond the expertise the authors, and so we have not attempted to cite relevant work from it.

[8] See Blanchette's (2001) and Shapiro's (2005) for an introduction.

[9] For example, we say in section 1.2.2 that formal languages can be thought of as models of natural languages; and much of the discussion on identity in section 9.3 raises issues as well.

Acknowledgments

This book began as a series of a lecture handouts written for the yearly mathematical logic class in the philosophy department at Rice University. The handouts were primarily written by Richard Grandy from 2007 to 2011, with both Stan Husi and Jacob Mills (TAs for the class during that time) contributing. Michael Barkasi began the process of typesetting these handouts in LaTeX for the fall of 2012 (at which time he was the TA). By next summer he finished this process, and over the next year he compiled and reorganized the handouts into a draft textbook, adding in material as needed. Since then Joshua Reagan (TA from 2014-17) and Richard Grandy have revised the text and added new material.

As with most other textbooks, no theorems are original and their presentation is, for the most part, what you'll find elsewhere. The proofs for all the theorems—along with the actual writing and presentation—have been done from scratch by one or more of the authors; but the general methods and approaches used are just those the authors learned from others. Thus this textbook owes just as much to previous work in logic as most other modern textbooks. It passes on to students a body of results and a general conceptual framework developed by many logicians over the years. It does so in roughly the style and presentation in which those results were passed to the authors themselves, with some changes they think are improvements. We note these as they come up.

A few more specific acknowledgments are called for. At the time the first author was preparing the original handouts he was using Merrie Bergmann, James Moor, and Jack Nelson's textbook *The Logic Book* (2003); hence some of the terminology and organization in this book reflects that. Benson Mates' classic *Elementary Logic* (1972), the previous text, and Church's *Introduction to Elementary Logic* (1956) also influenced the first author's approach. The second author has relied heavily on Wilfred Hodges's rich survey article "Elementary Predicate Logic" (2001b) for historical citations. The name "Dragnet" for theorem 4.16 comes from Michael Smith, another former TA for the first author. The problems in exercise 5.3.3 (on page 120) comes from Howard Pospesel & David Marans, *Arguments: Deductive Logic Exercises* (1978), which Pospesel and Marans have generously released to the public.

Chapter 1
Introduction

1.1 What is Logic?

1.1.1 Rational Creatures

Humans are rational creatures. As such, we reason about our circumstances and the world in general so we may understand them better. An improved understanding of the world helps us make informed decisions, which (ideally) tend to bring about preferable outcomes. If you find yourself lost in a labyrinth and hunted by a hungry minotaur, clever reasoning could help you escape. Poor reasoning could get you eaten!

At least some of our reasoning is *discursive*. Discursive reasoning is an iterative process: start with a set of initial claims and then infer a result from them, perhaps repeating the process until a certain conclusion is reached. Such a chain of inferences can be written down or otherwise recorded as an argument.

Knowledge of logic helps us exploit a particularly reliable kind of discursive reasoning. It enables us to judge an argument from any source according to independent, principled criteria. Not only can we assess the strength of arguments presented to us; we can construct rigorous arguments of our own to share with others. Rationality thus has an important social aspect. By sharing arguments with each other we can, to some extent, coordinate beliefs and behaviors, and thereby complete tasks beyond the ability of any one person.

The social role of argumentation provides a clue about the subject matter of logic. Arguments are shared by way of language. One can give an argument by writing it on paper, typing it in an e-mail, stating it in a speech, etc. Once inscribed or encoded in one linguistic medium or another, it can then be assessed by others. Accordingly, logicians must attend to certain public features of language.

1.1.2 Logical Consequence

Logic is the study of *logical consequence* and *logical truth*. When we say that one sentence 'logically follows' from another, we mean that the first sentence is a logical consequence of the other. But how can we identify when some claim is a logical consequence of another? As suggested previously, we must attend to certain features of the language in which the argument is given.

One sentence ψ is a logical consequence of another sentence ϕ if and only if the truth of ϕ guarantees, in virtue of its logical structure, the truth of ψ.

Three points are worth making about this provisional definition. First, don't be afraid of the Greek letters: they're just variables for sentences. Second, the phrase 'if and only if' is one we use throughout the text, sometimes abbreviated as 'iff'. 'ϕ if and only if ψ' is equivalent to 'If ϕ then ψ and if ψ then ϕ'. Third, a solid understanding of this definition depends on having a clear notion of *logical structure*. One of the central goals of this textbook is to provide such a notion.

Consider the following two sentences:

1. George Washington was in the Continental Army and Nathanael Green was in the Continental Army.
2. Nathanael Green was in the Continental Army.

The second sentence is a logical consequence of the first. If the first is true the second must be too. The next two sentences share the same structure as the last two:

3. Benjamin Franklin was a delegate to the Continental Congress and Thomas Jefferson was a delegate to the Continental Congress.
4. Thomas Jefferson was a delegate to the Continental Congress.

As above, the second sentence is a logical consequence of the first. We can show the common logical structure of the preceding pairs of sentences using a *schema*:

5. ϕ and ψ.
6. ψ.

We have replaced the non-logical content of each sentence with Greek letters and kept the logical word 'and'. The Greek letters serve as variables which stand for clauses of the English language. Any substitution of appropriate English clauses for the letters ϕ and ψ in the above schema generates a pair of sentences in which the latter follows from the former.[1]

A sentence can also be the logical consequence of a set of sentences. For example, given the sentences,

7. The sun is shining.
8. The birds are chirping.

the following is a logical consequence:

9. The sun is shining and the birds are chirping.

We can represent the logical structure of these sentences in schematic form:

[1] We say more about what counts as an appropriate substitution later in the chapter.

10. ϕ.
11. ψ.
12. ϕ and ψ.

Any substitution of appropriate English clauses for the letters ϕ and ψ in this schema generates three sentences in which the last follows logically from the first two.

1.1.3 Logical Truth

A sentence is a *logical truth* if and only if the logical structure of the sentence guarantees its truth. For example,

13. If the sun is shining then the sun is shining.

This sentence must be true, regardless of whether the sun is shining. Contrast that with sentence 8; if no birds exist, then 8 can't be true. There is nothing special about the non-logical content of 13. We could replace 'the sun is shining' with 'it's raining outside' and get another logical truth:

14. If it's raining outside then it's raining outside.

The schema of these two logical truths is:

15. If ϕ then ϕ.

The non-logical content is replaced with ϕ and the words 'if…then' are retained. Any substitution of an appropriate clause of English for ϕ generates a logical truth. More generally, logically true sentences have a structure such that, whatever appropriate clause(s) we substitute for the non-logical parts, the result is also a logical truth.

To construct general schemas from particular examples, as we did above, requires distinguishing between the logical structure and the non-logical content of a sentence. We take for granted that certain "logical" words play a special role in constituting the logical structure of an English sentence—e.g., words such as 'and', 'or', 'if…then', 'not', 'all', 'some', among many others.

Logical consequence and logical truth are closely related concepts. In fact, each can be defined in terms of the other, though we won't specify the connection until we give a mathematically precise definition of each. Unfortunately, natural languages such as English have features that make these concepts difficult (or even impossible) to define adequately.

1.2 Natural and Formal Languages

1.2.1 Natural Languages

Calling English a *natural* language is only meant to indicate that it developed gradually and informally over time, and wasn't the product of, say, scholars officially defining the

grammar and vocabulary from scratch. English, Greek, German, Russian, Spanish, Mandarin, and Hindi are all natural in this sense.

Let's examine some of the obstacles to achieving mathematical precision in natural language. First, it is indeterminate which collections of words and letters are genuine sentences. The following strings are, perhaps, neither clearly sentences nor clearly not sentences of English:[2]

16. Colorless green ideas sleep furiously.
17. Furiously sleep ideas green colorless.
18. Seventeen is purple.
19. Green is a prime number.
20. Someone someone admires admires someone.

There are no principled, universally accepted rules that determine whether each is officially a sentence of English. Without a clear distinction between sentences and non-sentences, we cannot be certain which are appropriate for substitution in logical schemas like those given in the previous section.

A second problem with natural languages is that they contain paradoxical sentences. A *paradoxical* sentence is one that's true if and only if it's false. The most famous examples are those that engender the liar paradox, often called *liar* sentences.[3] The following are two examples.

21. Sentence 21 is false.
22. The second sentence on this page with exactly twelve words is false.

Paradoxical sentences seem to be simultaneously both true and false. Try assuming that 21 is false, and it seems to follow that it's also true. Try assuming that 21 is true, and it comes out false. Both outcomes are impossible because they're each contradictory. Paradoxical sentences frustrate systematic and coherent logical analysis, so logicians prefer to exclude them altogether.

A third problem is that natural languages contain *ungrounded* sentences. The following pair is a good example:

23. Sentence 24 is true.
24. Sentence 23 is true.

These sentences seem to be unhinged from reality. We can assume consistently that both are true, and we can assume consistently that both are false, regardless of any substantive fact about the world. Most logicians prefer to exclude ungrounded sentences along with the paradoxical ones.

[2]Sentences 16 and 17 are from Chomsky 2002 [1957].

[3]See the following: Tarski 1983 [1933], 1944, Kripke 1975, Barwise and Etchemendy 1987, and Gupta 2001.

An important note is in order. Paradoxical and ungrounded sentences are different from *contradictory* (or "self-contradictory") ones. The following is a contradiction, not a paradox:

25. The door is open and it isn't open.

Unlike a paradoxical sentence, which seems to take both truth values, a contradictory sentence must be false. There is nothing wrong with contradictory sentences from a logician's point of view. They are perfectly legitimate relatives of logical truths. Just as a logical truth must be true, a contradiction (i.e., a logical falsehood) must be false. Contradictory sentences are self-refuting, but they take a single truth value and so are acceptable for our purposes.

A fourth problem with natural language is that many, if not most, sentences of natural languages have ambiguous or context-dependent meanings. Consider the following.

26. Wealthy Americans flee south in record numbers.
27. We gave the bananas to the monkeys because they were hungry.
28. We gave the bananas to the monkeys because they were tasty.
29. We gave the bananas to the monkeys because they were there.
30. Paul insulted George because he thought he was someone else.
31. Sally's car is red.
32. He is tall.
33. He is the tallest one here.
34. You are tall.
35. Sally went to the bank.

We can often resolve ambiguous sentence meaning by considering the particular circumstances in which a sentence is spoken or written. For instance, the ambiguity may come from an indexical term like 'you' (e.g., 34), in which case the intended referent is clear once we figure out who is speaking or writing to whom. Sometimes the meaning of a term is context-dependent, as with 'tall' (e.g., 32). Five feet would be tall for a seven-year old, but not for an adult. Other cases are more complicated. Does 'Sally's car' in 31 mean the car Sally owns, leases, or something else? Is 'red' the color of the car's exterior or interior? Is it light red, dark red, or something in between?

Pronoun reference can be difficult to resolve, even in context (e.g., 'he' in 30, 32, 33). Finally, sometimes a word in the sentence has multiple distinct meanings (e.g., 'bank' in 35) or the overall syntax of the sentence is ambiguous (e.g., 26–29). As a result the intended meaning of the sentence may not be recoverable from the context without additional information.

Ambiguous sentences can make careful logical analysis difficult or impossible. The following looks like a logical truth:

36. If she is tall then she is tall.

But what if the first 'she' refers to one person and the second refers to another?

1.2.2 Formal Languages

The logician avoids these difficulties by using carefully constructed formal languages.[4] A *formal* language has the following features:

(1) There is an explicit list of permitted symbols.

(2) There is a definition of which strings of symbols count as sentences.

A *string* of symbols is just a row or sequence of symbols. For example, 'fq3H7' is a string consisting of 'f' followed by 'q', '3', 'H', and '7'. The order of the characters is important; 'f7q' is not the same string as '7fq'.

Formal languages have a precisely defined syntax, but no fixed semantics. Definitions of sentences in formal languages make no reference to meanings of the sentences or their parts. These definitions make reference only to the *form*, or shape of the symbols and their arrangements in strings. We should think of the symbols and sentences of a formal language as lacking inherent meaning.

In a well-constructed formal language we can check whether any given string is a genuine sentence, solving the indeterminacy problem faced by natural languages. We also exclude the features of natural languages that lead to paradoxical and ungrounded sentences. Sentences of formal languages have no inherent meaning, but there are ways to assign meaning to them while avoiding the difficulties of natural language. In chapter 5 we provide methods of translating English sentences into constructed formal languages.

By turning to formal languages we are using a familiar (and often successful) strategy of replacing a hard messy problem with a more tractable mathematical one. For example, geometry doesn't deal with points or lines in the physical world; the points and lines of geometry have zero area and zero width, respectively. Nevertheless, geometry provides useful models of the physical world when physical points and lines approximate geometrical ones sufficiently. When the logical features of our formal language sufficiently reflect the logical features of our natural language, mathematical idealization is a productive strategy.[5]

How do our formal languages relate to natural languages like English? We think of the formal languages as abstract models of important parts of their natural counterparts. We cover several formal languages in this text: SL (chapter 2), QL1 (chapter 3), QL (chapter 4), and later MSL and QLI (chapter 9), will serve as progressively

[4] Poorly (or deviously) constructed formal languages can have the same problems as natural ones.

[5] Historically the move to formal languages in the study of logic (and mathematics) has been fruitful. One example, important for the development of modern logic, comes from set theory. By using formal methods logicians found that the naïve axioms of set theory are inconsistent (see Demopoulos and Clark 2005 for a quick overview of Russell's paradox, or Smullyan and Fitting 2010: ch 1 for a more careful discussion).

more detailed models of English. After learning how to define these formal languages and how to translate between them and English, the logic student should have a better understanding of the logical structure of English and of logical consequence more generally. The student should keep in mind that these formal languages cannot perfectly capture all the desirable logical features of English. There is a limit to both the range of English sentences they model and the accuracy with which they model them.

1.3 Some Distinctions

1.3.1 Use and Mention

There are three distinctions that are important in the modern study of logic. The first is the use/mention distinction. For any word or expression (string of words), we can either *use* the word or expression, or instead *mention* it. *Using* words and expressions is what we normally do, while *mentioning* a word or expression is one way to talk about that word or expression. For example, you could say (i) that a cow is a four-legged bovine, or you could instead say (ii) that 'cow' has three letters. In case (i) we are talking about a particular kind of animal, while in case (ii) we are talking about the *word* for that animal. In this textbook words that are mentioned are put in single quotes. Here are some examples:

37. Houston has over 2 million inhabitants. [True]
38. Houston has over 10 million inhabitants. [False]
39. 'Houston' has more than 2 million inhabitants. [False]
40. 'Houston' has more than 4 letters. [True]
41. Austin has more than 4 letters. [False]
42. 'Austin' has exactly 6 letters. [True]

1.3.2 Type and Token

The next distinction is the type/token distinction. How many letters does 'Houston' have? The answer depends on whether the asker means letter tokens or letter types. A *token* is a physical object or event, while a *type* is an abstract *kind* of physical object or event. Tokens are located in space and time, while types presumably are not. The word 'Houston' has 7 letter tokens and 6 letter types, because the letter 'o' occurs twice. In the sentence 'Ron went on and on' there are five word tokens and four word types.[6] In 'radar' there are five letter tokens and three letter types.

[6] In the sentence 'Ron went on and on', there are three occurrences of the string 'on' with one of them in the name 'Ron'. However, this instance doesn't count as a word token in English, because it's only a piece of a word, not a complete word. Other cases in English are more ambiguous. In the compound word 'footnote' does 'note' count as a word token?

1.3.3 Object Language and Metalanguage

The third distinction is that between an object language and a metalanguage. Returning to the use/mention distinction, if one mentions a word, then the *object language* is the language of the word being mentioned and the *metalanguage* is the language in which that word is mentioned. For example, if I say "The German word for dog is 'Hund'," I've used English (the metalanguage) to talk about German (the object language). In this text the metalanguage will always be English augmented with some carefully selected mathematical notation. The object language will be a formal language.

Usually there's a little more to being a metalanguage than the mere fact that it's the language used to talk about another. Often there's some specific purpose. For example, the metalanguage might be used to give definitions of words in the object language. Or, as in our case, the metalanguage might be used to define when a sentence of the object language is true, or to describe when the logical consequence relationship holds between sentences of the object language.

1.4 MathEnglish

The various formal languages in this text are to be object languages. However, before we can define our first formal language (SL) we need to familiarize the reader with certain mathematical concepts. The metalanguage contains a mix of English, mathematical symbols (e.g., set theory operators), and other technical jargon. We call our metalanguage *MathEnglish*.

1.4.1 Metavariables

One important tool for MathEnglish is the *metavariable*, which we define as a variable for strings of the object language. We use lowercase Greek letters as our metavariables, as we did in the schemas given at the beginning of the chapter. The three we use most commonly are 'ϕ' (phi), 'θ' (theta), and 'ψ' (psi). These Greek letters are not in the object language, but instead are used in the metalanguage for talking about the object language.

An example will make their role more clear. Let's say that MathEnglish is our metalanguage and (regular) English is our object language.[7] Recall the logically true sentence from the beginning of the chapter:

> If the sun is shining then the sun is shining.

We can replace 'the sun is shining' with any other assertion and get another logically true sentence. With this in mind, consider the following claim of MathEnglish:

> All English sentences of the form 'If ϕ then ϕ' are logical truths.

[7] As discussed earlier, in later chapters we use a formal language and not English as our object language, but for the purpose of illustration we temporarily ignore the problems of natural language.

The use of 'ϕ' helps us talk about *all* English sentences of a certain form as logically true, or at least all English sentences of a certain kind. As logicians we are concerned with truth and truth-preservation, so we are only going to address strings with plausible truth values, i.e. declarative sentences. A declarative sentence is one that makes an assertion. By restricting the substitutions to declaratives, we rule out substitutions such as 'Ergle bergle barfle narfle,' 'nef34fwjh9ufh32,' 'Kentucky fried chicken,' and 'Are you cooking?'. We cannot legitimately affirm truth or falsity of these non-assertions. In the rest of the text, when we refer to 'all' sentences of a natural language we only mean the declarative ones.

In any case, metavariables are indispensable tools for making general claims about object languages.

1.4.2 Sets

MathEnglish makes use of set theory. A *set* is just a collection of items called *elements*. Elements can be anything, like animals, people, planets, cartoon characters, or even abstract objects such as numbers. A set can be specified using curly brackets, as in the following set of integers from 1 to 4:

$$\{1, 2, 3, 4\}$$

The order of elements in set notation doesn't matter, so the following set is identical to the last:

$$\{2, 3, 1, 4\}$$

Furthermore, repetition of an element in set notation is to be disregarded; an element can only be in the set once. Accordingly, the following sets are the same set as the last two:

$$\{1, 2, 2, 3, 3, 3, 4, 4, 4, 4\}$$
$$\{2, 2, 3, 1, 4, 2, 3, 1, 4, 4, 3\}$$

Sets don't have to be finite. Consider the set of positive integers:

$$\{1, 2, 3, 4, ..., 1002, 1003, 1004, ...\}$$

There is one set that doesn't have any elements, and it's called either the *null set* or the *empty set*. Symbolically the empty set is denoted by either $\{\}$ or $\emptyset$.

Set membership is expressed with the '$\in$' symbol. Let the Greek letters Δ (delta) and Γ (gamma) stand for arbitrary sets. We assert that 7 is an element of set Δ and 3 is an element of Γ as follows: $7 \in \Delta$, $3 \in \Gamma$. On this notation,

$$2 \in \{1, 2, 3\}$$

is true and,

$$4 \in \{1, 2, 3\}$$

is false.

We use lowercase Greek letters α (alpha) and β (beta) as metavariables for elements. For example, if α and β are odd numbers and Δ is the set of even numbers, then $\alpha + \beta \in \Delta$.

Definition 1.1. For any two sets Δ and Γ, Δ is a *subset* of Γ iff for each α such that $\alpha \in \Delta$, $\alpha \in \Gamma$.

For example, $\{1, 2, 3\}$ is a subset of $\{1, 2, 3, 4\}$. The set $\{1, 2\}$ is a subset of each of the two previous sets. The symbol '$\subseteq$' denotes the subset relation. So if Δ is a subset of Γ, $\Delta \subseteq \Gamma$. Note that by the above definition every set is a subset of itself.

Definition 1.2. For any two sets Δ and Γ, Δ is a *proper subset* of Γ iff

(1) $\Delta \subseteq \Gamma$ and

(2) there is some α such that $\alpha \notin \Delta$ and $\alpha \in \Gamma$.

As you can probably guess, the '$\notin$' symbol means *is not an element of*. The symbolic notation for 'Δ is a proper subset of Γ' is $\Delta \subset \Gamma$. No set is a proper subset of itself.

Take care to avoid confusing the membership, subset, and proper subset relations.

43. $2 \in \{1, 2, 3\}$ [True]
44. $2 \subseteq \{1, 2, 3\}$ [False]
45. $\{2, 3\} \subseteq \{1, 2, 3\}$ [True]
46. $\{2, 3\} \in \{1, 2, 3\}$ [False]
47. $\{2, 3\} \subset \{1, 2, 3\}$ [True]
48. $\{1, 2, 3\} \subset \{1, 2, 3\}$ [False]

1.4.3 More on Sets: Union and Intersection

Sometimes a set is defined by reference to other sets. Say that there is a set Δ that contains all the members of two other sets, Γ_1 and Γ_2, and nothing else. Then Δ is said to be the *union* of Γ_1 and Γ_2. Let's give a strict definition:

Definition 1.3. $\Delta = \Gamma_1 \cup \Gamma_2$ iff

(1) for each $\alpha \in \Gamma_1$, $\alpha \in \Delta$ and

(2) for each $\alpha \in \Gamma_2$, $\alpha \in \Delta$ and

(3) for each $\alpha \in \Delta$, $\alpha \in \Gamma_1$ or $\alpha \in \Gamma_2$ (or both).

The symbol '$\cup$' means *union*. Some examples:

$$\{\text{Mercury, Jupiter, Mars, Saturn}\} = \{\text{Mercury, Jupiter}\} \cup \{\text{Mars, Saturn}\}.$$

$$\{\text{Newton, Bacon, Boyle}\} = \{\text{Newton, Bacon}\} \cup \{\text{Bacon, Boyle}\}.$$

Another useful concept is *intersection*. The set Δ is the intersection of two sets Γ_1 and Γ_2 iff Δ contains all the elements shared by both Γ_1 and Γ_2 but nothing else. More strictly:

Definition 1.4. $\Delta = \Gamma_1 \cap \Gamma_2$ iff

(1) for each α such that $\alpha \in \Gamma_1$ and $\alpha \in \Gamma_2$, $\alpha \in \Delta$ and

(2) for each $\alpha \in \Delta$, $\alpha \in \Gamma_1$ and $\alpha \in \Gamma_2$.

The symbol '$\cap$' means *intersection*. The intersection of {Mercury, Jupiter} and {Mars, Saturn} is the empty set, {}, because the former two sets have no members in common.

$$\{\text{Mercury, Jupiter}\} \cap \{\text{Mars, Saturn}\} = \{\}.$$

By contrast, the intersection of {1, 5, 6, 9} and {2, 3, 5, 6} is {5, 6}.

$$\{1, 5, 6, 9\} \cap \{2, 3, 5, 6\} = \{5, 6\}.$$

Another example:

$$\{\text{Carnap, Quine}\} = \{\text{Anscombe, Quine, Carnap}\} \cap \{\text{Carnap, Lewis, Quine}\}.$$

1.4.4 Ordered Pairs and Ordered n-tuples

Sometimes we would like to describe a collection in which, unlike sets, the order does matter. For that purpose we have *ordered pairs* and *ordered n-tuples*. An ordered pair is a two-place collection in which the order matters. To differentiate ordered pairs from sets, we indicate them using angled brackets rather than curly brackets. So, while the sets {1, 2} and {2, 1} are identical, the ordered pairs ⟨1, 2⟩ and ⟨2, 1⟩ are not.

An ordered collection having more than two places is an n-tuple, where n is the number of places needed. So, ⟨Mercury, Venus, Earth⟩ is a 3-tuple, ⟨Jupiter, Saturn, Uranus, Neptune⟩ is a 4-tuple, and so on.

Also unlike sets, for ordered pairs and n-tuples, repetition makes a difference. For example, $\langle 1, 2, 3\rangle$ is different from $\langle 1, 2, 3, 3, 3\rangle$.

1.4.5 Mathematical Proofs

We use proofs to establish that something has certain mathematical properties. A proof works from one or more definitions to some interesting result, called a *theorem*. Much of this textbook is filled with proofs of theorems about the various formal languages defined in later chapters. If we want to prove an intermediate result that

isn't by itself particularly interesting but which is handy for use in one or more other proofs, we sometimes call it a *lemma*.

To illustrate, let's look at a simple proof.

Theorem 1.5. Let Δ_1, Δ_2, and Δ_3 be sets. If $\Delta_1 \subseteq \Delta_2$ and $\Delta_2 \subseteq \Delta_3$ then $\Delta_1 \subseteq \Delta_3$.

Proof: Assume that $\Delta_1 \subseteq \Delta_2$ and $\Delta_2 \subseteq \Delta_3$. Since $\Delta_1 \subseteq \Delta_2$, then by the definition of $\subseteq$, any α in Δ_1 is also in Δ_2. And since $\Delta_2 \subseteq \Delta_3$, any α in Δ_2 is also in Δ_3. It then follows that if $\alpha \in \Delta_1$ then $\alpha \in \Delta_3$. Thus, by the definition of $\subseteq$, $\Delta_1 \subseteq \Delta_3$. ■

> Some students are not used to the structure and rigor of mathematical proofs. As an aid to understanding we sometimes include commentary to explain how a particular proof works. We use a grayed box to signify that this commentary is not itself part of the proof.
>
> Here we are trying to prove a *conditional* theorem, i.e., a statement of the form "If A then B." The typical way to demonstrate such theorems is to start by assuming the antecedent, A. Then we work from that assumption to show that the consequent, B, must follow. Each step is justified by reference to a given definition or by mathematical reasoning, often in the language of set theory.

We have proved that '$\subseteq$' is transitive. Having achieved this result, we permit ourselves to use it throughout the rest of the text without proving it again. The end of a completed proof is indicated with a black box: ■.

1.4.6 Recursive Definitions

In later chapters we use MathEnglish to define concepts such as *sentence*, *truth*, and *entailment* for a given object language. Many of these definitions are recursive. Recursive definitions are an invaluable tool for logic because they enable us to give a rigorous finite definition of an infinite set. They can be understood as defining which elements are in a given set.

Definition 1.6. A *recursive definition* of Δ has three clauses:

(1) the *base clause(s)*, which specifies one or more objects as elements of Δ,

(2) the *generating clause(s)*, which specifies one or more ways of generating (or finding) other objects that are elements of Δ, and

(3) the *closure clause*, which specifies that something is in Δ only if it can be shown to be so by applications of the first two clauses.

Think of the base clause as the place to specify specific elements for inclusion in the set. The generating clause is more powerful. It defines elements of the set by reference to more basic elements already included. The closure clause is usually a simple statement, but it is essential to specify what isn't in the set.

One concept that we can define recursively is *natural number*.

Definition 1.7. The set $\mathbb{N}$ of natural numbers:

(1) Base Clause: Let 0 be a natural number. I.e. $0 \in \mathbb{N}$.

(2) Generating Clause: If $n \in \mathbb{N}$ then $n + 1 \in \mathbb{N}$.

(3) Closure Clause: Nothing else is in $\mathbb{N}$.

This definition unambiguously identifies infinitely many numbers as natural numbers. Obviously 0 is a natural number. Is 7 a natural number? It is. We know 0 is a natural number because the base clause says so. Thus, according to the generating clause, $0 + 1$ (i.e., 1) is also a natural number. Apply the generating clause again to show that $1 + 1$ (i.e., 2) is natural number. One can continue applying the generating clause until 7 is reached. Is -3 a natural number? No. -3 is less than 0 and the generating clause only defines larger numbers as natural. The closure clause rules out the inclusion of anything else.

Let's use a recursive definition to identify which people are ancestors of French mathematician Blaise Pascal.

(1) Base Clause: Blaise Pascal's mother and father are ancestors of Blaise Pascal.

(2) Generating Clause: If x is an ancestor of Blaise Pascal and y is a parent of x, then y is also an ancestor of Blaise Pascal.

(3) Closure Clause: No one else is an ancestor of Blaise Pascal.

This definition picks out the mother and father of Blaise Pascal, their respective mothers and fathers, the mothers and fathers of the latter, and so on. It excludes everyone else. Recursive definitions are powerful, because in a few lines we can fix an unambiguous collection that is arbitrarily large. The closure clause is usually relatively trivial (e.g., "Nothing else is an x.").

1.4.7 Conclusion

Now we have most of the mathematical tools we need to define our first formal language, *Sentential Logic*. Anything else we need will be given along the way.

1.5 Exercises

1.5.1 Sets

1. True or false: $\{2, 3\} \subseteq \{1, 2, 3\}$
2. True or false: $\{1, 2, 3\} \subset \{1, 2, 3\}$
3. True or false: $\{2, 3\} \in \{1, 2, 3\}$

4. True or false: $\{2,3\} \in \{1,\{2,3\}\}$
5. True or false: $2 \in \{1,\{2,3\}\}$
6. True or false: $\{1,2,3\} = \{1,\{2,3\}\}$
7. True or false: $\{1,2,3\} = \{3,3,1,2,3,2,2,1\}$

1.5.2 Unions and Intersections

1. True or false: $\{2,4,6\} \cup \{1,2,3\} \subseteq \{1,2,3,4,5,6\}$
2. True or false: $\{2,4,6\} \cap \{1,2,3\} = \{1,2,3,4,5,6\}$
3. True or false: $\{1,2\} \cap \{3,4\} \subseteq \{1,2\} \cup \{3,4\}$
4. True or false: $\{1\} \cup \{2,3\} = \{1,\{2,3\}\}$

1.5.3 Proofs

Prove each of the following.

1. $\Delta_1 \cap \Delta_2 \subseteq \Delta_1 \cup \Delta_2$.
2. If $\Delta_1 \subseteq \Delta_2$ then $\Delta_1 \cup \Delta_3 \subseteq \Delta_2 \cup \Delta_3$.
3. If $\Delta_1 \subseteq \Delta_2$ then $\Delta_1 \cap \Delta_3 \subseteq \Delta_2 \cap \Delta_3$.

1.5.4 Recursive Definitions

Define the following recursively.

1. The ancestors of George Washington.
2. All positive even numbers.
3. All integers.

Chapter 2
Sentential Logic

2.1 The Language SL

2.1.1 Sentences of SL

Our first task is to define the syntax of the basic formal language SL.[1] This language is variously called *sentential logic* or *propositional logic*. We use the former name because it's easier to get a grasp on what a sentence is. Our choice allows us to sidestep the philosophical debate over what a proposition is.[2]

As stated in the previous chapter, each formal language requires (1) a list of basic symbols, and (2) a specification of which sequences of those symbols count as sentences. These determine the proper grammar or syntax of the language.

Definition 2.1. The *basic symbols* of SL are of three kinds:[3]

(1) Logical Connectives: $\sim$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

(2) Punctuation Symbols: (,)

(3) Sentence Letters: $A, B, \ldots, T, A_1, B_1, \ldots, T_1, A_2, B_2, \ldots$

Logical connectives are sometimes called logical symbols, logical operators, logical terms, or logical functors.[4] One could draw important distinctions between symbols, terms, and operators, but the names are just as often used interchangeably. The *sentence letters* are italicized capital letters of the Roman alphabet from 'A' to 'T'. To give ourselves an infinite supply, any of these letters with a subscripted positive integer is also a sentence letter. E.g. C and a subscripted 7 can be combined to get the new sentence letter C_7. Remember that C_7 and C are different sentences.

[1] Work on modern sentential logic originated with George Boole (1854) and Augustus De Morgan (1847; 1860). As Christine Ladd-Franklin notes (1883: 17) before giving her own, variations quickly followed from William S. Jevons, Ernst Schröder, Hugh McColl, and Charles S. Peirce.

[2] For historical and contemporary discussions of propositions, see Frege 1892, Russell 1996 [1903]: 13,47, Church 1956: 26, Quine 1986: ch. 1, Schiffer 1987: ch. 3, Grandy 1993, Bealer 1998, King 2007, Soames 2010.

[3] The commas and ellipses are *not* symbols of SL.

[4] In other textbooks there are sometimes different symbols used for these connectives. Along with $\sim$, $\neg$ and $-$ are used for negation, & and $\cdot$ for conjunction, $\supset$ and $\Rightarrow$ for conditionals, and $\equiv$ for biconditionals. $\vee$ is almost universally the symbol for disjunction.

The following list gives the name of each connective symbol:

(1) $\sim$: "tilde",
(2) $\wedge$: "wedge",
(3) $\vee$: "vee",
(4) $\rightarrow$: "arrow",
(5) $\leftrightarrow$: "double-arrow".

Definition 2.2. The *sentences of SL* are given by the following recursive definition:

Base Clause: Every sentence letter is a sentence.

Generating Clauses:

(1) If ϕ is a sentence, then so is $\sim\phi$.[5]
(2) If ϕ and θ are sentences, then so are both $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.
(3) If all of $\phi_1, \phi_2, \ldots, \phi_n$ are sentences (where n is an integer ≥ 2), then so are $(\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n)$.

Closure Clause: No other string is an SL sentence.

Here are some example SL sentences:

1. B
2. $\sim B$
3. $(\sim B \rightarrow C)$
4. $\sim(\sim B \leftrightarrow C)$
5. $(A \vee C \vee D)$
6. $((A \rightarrow E) \vee C \vee \sim D \vee G)$
7. $((A \vee B) \wedge (C \vee D))$
8. $\sim(A \wedge \sim(B \rightarrow C))$

Sentence letters are sometimes called *atomic* sentences. Sentences of the form $\sim\phi$ are *negations*. Sentences of the form $(\phi \rightarrow \theta)$ are *conditionals*, and those of the form $(\phi \leftrightarrow \theta)$ are *biconditionals*. The left-hand side of the conditional, ϕ, is traditionally called the *antecedent* and the right-hand side, θ, the *consequent*. We often use the alternative terminology LHS (left-hand side) and RHS (right-hand side). Sentences of the form $(\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n)$ are *conjunctions* and their component sentences (e.g. ϕ_1) are *conjuncts*. Sentences of the form $(\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n)$ are *disjunctions* and their component sentences are *disjuncts*.[6]

[5] Remember from Chapter 1 that ϕ and θ are used as metavariables. In this definition they stand for sentences of SL.

[6] Many logic books treat conjunction and disjunction as binary (2-place), e.g., '$(A \vee B)$'. According to our definition '$(A \vee B \vee C)$' is also a perfectly good sentence. Textbooks that treat disjunctions as binary require an extra set of parentheses to generate an equivalent sentence: e.g., '$((A \vee B) \vee C)$'. This is also an acceptable sentence of SL, but the extra parentheses don't add interesting information. We prefer the definitions of conjunction and disjunction given above because they're closer to natural English and they allow us to avoid unnecessary parentheses.

The base and generating clauses tell us which strings are sentences, but they don't say which strings aren't. How do we know, for example, that '$(B(\rightarrow A$' isn't an SL sentence? That's what the closure clause is for. It explicitly excludes strings that cannot be constructed with the base and generating clauses.

Note that $\sim\phi$, $(\phi\rightarrow\theta)$, etc. in the generating clauses are **not SL sentences**. That's because metavariables aren't included in the symbols of SL. These *sentence schemas* are strings that can be made into sentences by substituting sentences for metavariables. For example, the substitution ϕ = 'A', θ = '$(C\leftrightarrow D)$' in the schema $(\phi\rightarrow\theta)$ results in the sentence '$(A\rightarrow(C\leftrightarrow D))$'.[7]

For any sequence of SL symbols it is possible to *prove* whether it is a sentence.

Example 2.3. $((A\vee(D\rightarrow B))\wedge\sim G)$ is a sentence of SL.
The base clause of 2.2 defines each of A, D, B, and G as sentences. From D and B, and by generating clause (2), $(D\rightarrow B)$ is a sentence. From G and generating clause (1), $\sim G$ is a sentence. From A and $(D\rightarrow B)$, and generating clause (3), $(A\vee(D\rightarrow B))$ is a sentence. And, finally, from $(A\vee(D\rightarrow B))$ and $\sim G$, and generating clause (3), $((A\vee(D\rightarrow B))\wedge\sim G)$ is a sentence.

2.1.2 Official and Unofficial Sentences of SL

Definition 2.2 is the definition of an *official* sentence of SL. For convenience we often work with *un*official sentences.

Definition 2.4. A string of symbols is an *unofficial* sentence iff we can obtain it from an official sentence by

(1) deleting outer parentheses, or

(2) replacing one or more pairs of official round parentheses ( ) with square brackets [ ] or curly brackets { }.

Thus '$A\wedge B\wedge C$' is an unofficial sentence, as are

1. $\sim([A\wedge B]\vee[C\wedge D])$
2. $(A\wedge B)\wedge C$
3. $(A\wedge B)\rightarrow C$
4. $(A\wedge[B\rightarrow C])$
5. $\sim A\vee C$
6. $[(A\wedge B)\wedge C]$
7. $\{A\wedge B\}\wedge C$
8. $A\wedge[B\rightarrow C]$

From an unofficial sentence we can unambiguously reconstruct the related official sentence. Throughout the rest of this text we usually drop outer parentheses, but we consistently use the standard parentheses ( ). The reader should feel free to use brackets [ ] or curly parentheses { } as is helpful.

[7] When Greek letters 'ϕ', 'ψ', 'θ', etc. are used it should usually be assumed that they range over SL sentences and not mere strings of SL symbols. In exceptional cases we intend for them to range over all strings of SL symbols, as we did in the definition just given of SL sentences.

2.1.3 A Comment on Use and Mention

Most of the time when you see Greek letters in the text, as in definition 2.2 (on page 16), we are *using* them, not mentioning them. Thus it's appropriate that in definition 2.2 we did not put them in single quotes. By contrast, we generally *mention* official and unofficial SL sentences rather than use them. Because we mention SL sentences so often it would be tedious to put them in quotes. Therefore we refrain from doing so unless there is some special reason to do so. We are also less strict when mentioning the basic symbols of SL.[8]

2.1.4 Other Properties of Sentences

Next we define four important properties and related features of sentences: subsentence, order, main connective, and construction tree.

Definition 2.5. The following clauses define when one sentence is a *subsentence* of another:

(1) Every sentence is a subsentence of itself.

(2) ϕ is a subsentence of $\sim\phi$.

(3) ϕ and θ are subsentences of $(\phi\rightarrow\theta)$ and $(\phi\leftrightarrow\theta)$.

(4) Each of $\phi_1, \phi_2, \ldots, \phi_n$ is a subsentence of $(\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n)$.

(5) (Transitivity) If ϕ is a subsentence of θ and θ is a subsentence of ψ, then ϕ is a subsentence of ψ.

(6) That's all.

Example 2.6. The sentence $(B\rightarrow C)$ has 3 subsentences:

(1) $(B\rightarrow C)$

(2) B

(3) C

Subsentences are counted by token, not type. Hence the similar $(B\rightarrow B)$ also has three subsentences; the two tokens of B are counted separately.

Example 2.7. $(A \vee (D\rightarrow B)) \wedge \sim G$ has 8 subsentences:

(1) $(A \vee (D\rightarrow B)) \wedge \sim G$

(2) $A \vee (D\rightarrow B)$

(3) $D\rightarrow B$

(4) A

(5) D

(6) B

[8] This is the usual convention. See e.g. Hodges 2001b: 7.

(7) $\sim G$ (8) G

Definition 2.8. A sentence ϕ is a *proper subsentence* of ψ iff ϕ is a subsentence of but isn't identical to ψ.

Each sentence is a subsentence of itself, but no sentence is a proper subsentence of itself.

Definition 2.9. The following clauses define the *order* of every SL sentence.[9] Let ORDϕ be the order of ϕ. Then:

(1) If ϕ is an atomic sentence (a sentence letter), then ORD$\phi = 1$.

(2) For any sentence ϕ, ORD$\sim\phi$ = ORD$\phi + 1$.

(3) For any sentences ϕ and θ, ORD$(\phi \rightarrow \theta)$ is one greater than the max of ORDϕ and ORDθ. Likewise, ORD$(\phi \leftrightarrow \theta)$ is one greater than the max of ORDϕ and ORDθ.

(4) For any sentences $\phi_1, \ldots, \phi_n$, ORD$(\phi_1 \wedge \ldots \wedge \phi_n)$ is one greater than the max of ORDϕ_1, ..., ORDϕ_n.

(5) For any sentences $\phi_1, \ldots, \phi_n$, ORD$(\phi_1 \vee \ldots \vee \phi_n)$ is one greater than the max of ORDϕ_1, ..., ORDϕ_n.

(6) That's all.

Example 2.10. What is the order of $(A \vee (D \rightarrow B)) \wedge \sim G$? The order of an atomic sentence is 1, so A, D, B, G each have order 1. The order of $\phi \rightarrow \psi$ is 1 plus the maximum of the orders of ϕ and ψ; thus the order of $D \rightarrow B$ is 2. Because ORD$\sim\phi$ = ORD$\phi + 1$, ORD$(\sim G)$ = ORD$(G) + 1 = 2$. Because the order of a disjunction $\phi_1 \vee \ldots \vee \phi_n$ is 1 plus the maximum order of the disjuncts, the order of $A \vee (D \rightarrow B)$ is 3. The same goes for conjunctions, so the order of $(A \vee (D \rightarrow B)) \wedge \sim G$ is 4.

Definition 2.11. The *main connective* is the connective token (or tokens) that occur(s) in the sentence but in no proper subsentence.

Example 2.12. The main connective in each of the following sentences has been underlined.

(1) $(A \vee (D \rightarrow B)) \underline{\wedge} \sim G$

(2) $L \underline{\vee} K \underline{\vee} H$

(3) $L \underline{\vee} (A \rightarrow B) \underline{\vee} H$

(4) $\underline{\sim}(L \vee K \vee H)$

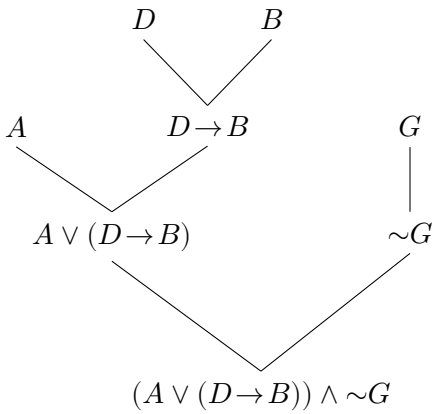
(5) $(((D \rightarrow E) \vee A) \underline{\wedge} (\sim B \wedge \sim (C \leftrightarrow H)))$

(6) $(\sim B \underline{\wedge} \sim (C \leftrightarrow H))$

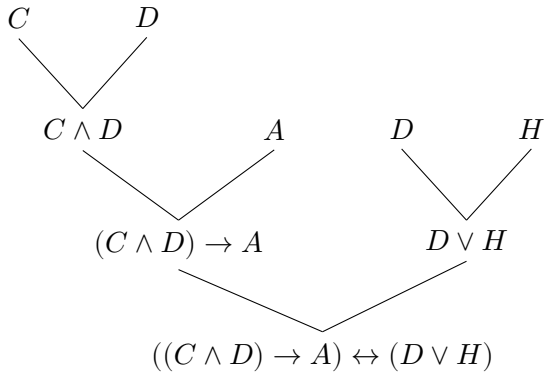
[9] (Post 1921, Hodges 2001b: 11)

Definition 2.13. The *construction tree* for a sentence is a diagram of how the sentence is generated through the recursive clauses of the definition of SL sentences. We put atomic sentences as leaves at the top, and the generating clauses specify how we can join nodes of the tree together (starting with the leaves at the top) into new nodes. The complete sentence is the node at the base of the tree.

Example 2.14. Give the construction tree for $(A \vee (D \rightarrow B)) \wedge \sim G$.



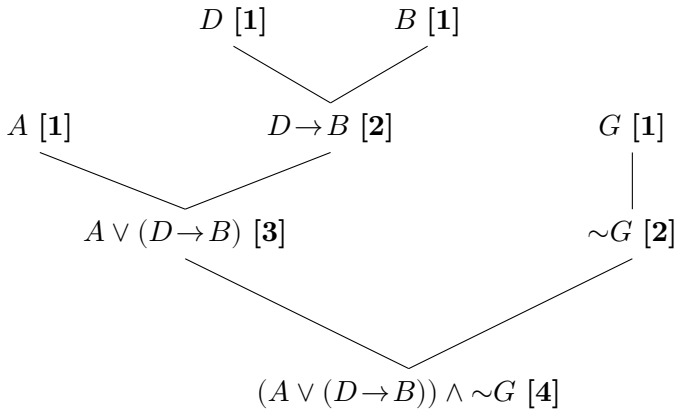
Example 2.15. Give the construction tree for $((C \wedge D) \rightarrow A) \leftrightarrow (D \vee H)$.



There are some helpful relationships between the construction tree of a sentence, its order, its subsentences, and its main connective. The subsentences of a sentence are the nodes in the sentence's construction tree. The order of a sentence is the number of nodes of its longest branch from root to leaf, i.e., the height of the tree. The main connective of a sentence is the connective added last (at the bottom) of the construction tree.

Example 2.16. Consider again the construction tree for $(A \vee (D \rightarrow B)) \wedge \sim G$. Find the order of the sentence by counting the height of the branches of the tree. (We count

up as we work our way down the branches.) Note that the answer we get agrees with that computed in example 2.10 (on page 19).



2.1.5 How Many SL Sentences are There?

There are at least as many sentence letters as there are natural numbers, and the sentence letters are a proper subset of the set of sentences. Are there more sentences than natural numbers? Below we prove there are not by matching up sentences with natural numbers.

Theorem 2.17. The number of SL sentences is equal to the number of natural numbers.

Proof: First, assign each sentence letter a natural number that only contains the digit '1', for example:

A	B	C	D	...
1	11	111	1111	...

Next, assign numbers to the other symbols of SL, for example:

$\sim$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	(	)
2	3	4	5	6	7	8

Given any sentence, replace its symbols with the associated numbers. For example, $\sim(A \wedge B \wedge \sim D)$ gets mapped to 2713113211118. For any sentence of SL, there is a unique natural number defined by this process. For any natural number we can determine if it represents an SL sentence, and if so which one. ■

This is not the most efficient way of representing sentences with numbers, but it is a simple one that avoids the use of special properties (e.g., being a prime number).

2.2 Models

SL is a formal language, so its sentences are mere strings of symbols without any inherent meaning—they don't "say anything" about the world.[10] As a consequence, sentences of SL lack an inherent truth value. That is, they are neither true nor false. Even so, nothing stops us from *assigning* truth values to SL sentences. In this section we explain how to do so consistently.[11]

2.2.1 Truth in a Model

Truth values are assigned to sentences by models.

Definition 2.18. A *model of an SL sentence* ϕ is an assignment of a truth value, either true or false, to each sentence letter in ϕ.

We can think of a model of ϕ as a function from the set of sentence letters of ϕ to the set of truth values: $\{\mathsf{T}, \mathsf{F}\}$. We use the letter '$\mathfrak{m}$' (a fraktur-style 'm') to represent such a function. If a model $\mathfrak{m}$ assigns a sentence letter ψ the value 'true', $\mathfrak{m}(\psi) = \mathsf{T}$. For the value 'false', $\mathfrak{m}(\psi) = \mathsf{F}$.[12]

To illustrate, a model for $(A \vee B \vee C)$ assigns a truth value to each of the sentence letters A, B, and C. Any model for $(A \vee B \vee C)$ is therefore also a model for $(A \wedge B \wedge C)$ and $((A \rightarrow B) \wedge \sim C)$; they all have the same sentence letters.

We often speak informally of *models* without making reference to any particular SL sentence. It's useful to talk this way because any given model of ϕ is a model of any other SL sentence with the same sentence letters, or a subset of them. If $\mathfrak{m}$ makes assignments to A, B, and C, then $\mathfrak{m}$ is a model of all the sentences that only contain sentence letters from that list. Accordingly, we define a model for a *set* of SL sentences.

Definition 2.19. $\mathfrak{m}$ is a *model of a set of sentences* Δ iff $\mathfrak{m}$ is a model for each sentence in Δ.

Consider a model $\mathfrak{m}$ that makes a truth value assignment to every sentence letter of SL. No sentence letter lacks an assignment, so it follows that $\mathfrak{m}$ is a model for the set of all SL sentences. The following definition characterizes such models as *models of SL*.

[10] We mentioned this property of formal languages in 1.2.2.

[11] You might ask why we don't first assign SL sentences meanings, then determine whether they are true or false based on those meanings. There are difficulties associated with assigning meanings that would complicate our project unnecessarily. (We show how to *interpret* SL sentences as having meanings in chapter 5.) One of the most important discoveries in logic was that for SL only truth values matter; all other details of meaning are irrelevant.

[12] The definiton of 'model' we use assigns one of two truth values to each sentence letter, but that's not the only way to define them. Other definitions assign more than two. The assumption that there are only two truth values simplifies analysis and is widely shared, but whether two is enough is a matter of philosophical debate. Nevertheless, even if our definition of a model is a simplification, it is a historically fruitful one and helps us better understand logical consequence. In chapter 9 we discuss formal languages with additional truth values.

Definition 2.20. $\mathfrak{m}$ is a *model of SL* iff $\mathfrak{m}$ is a model of every sentence of SL.[13]

Models can be uniform; for example, there is a model that assigns true to every sentence letter. Or they can be given according to an arbitrary pattern, such as the model that alternately assigns true or false to a list of sentence letters. A model can even assign truth values at random. *Every* possible function from sentence letters to truth values is a model.

A model of ϕ only assigns truth values to the sentence letters of ϕ. It does not directly assign a truth value to any of the complex (i.e. non-atomic) sentences of SL. The truth value of a complex sentence ϕ depends upon two things: (1) the main connective of ϕ, and (2) the truth values of the proper subsentences of ϕ. Each logical connective is associated with a truth function. While the truth assignments to the sentence letters vary by model, the truth functions of the connectives do not.[14]

Definition 2.21. The following clauses define whether an SL sentence θ is *true* or *false* on a model $\mathfrak{m}$ for θ. The relevant clause is determined by which main connective θ has, if any:

(1) θ is a sentence letter. θ is true on $\mathfrak{m}$ iff $\mathfrak{m}$ assigns true to it, i.e. $\mathfrak{m}(\theta) = \mathsf{T}$.

(2) θ is of the form $\sim\phi$. θ is true on $\mathfrak{m}$ iff ϕ is false on $\mathfrak{m}$.

(3) θ is of the form $(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$. θ is true on $\mathfrak{m}$ iff each of the conjuncts $\phi_1, \phi_2, ..., \phi_n$ is true on $\mathfrak{m}$.

(4) θ is of the form $(\phi_1 \vee \phi_2 \vee ... \vee \phi_n)$. θ is true on $\mathfrak{m}$ iff at least one of the disjuncts $\phi_1, \phi_2, ..., \phi_n$ is true on $\mathfrak{m}$.

(5) θ is of the form $\phi \rightarrow \psi$. θ is true on $\mathfrak{m}$ iff the LHS ϕ is false or the RHS ψ is true on $\mathfrak{m}$ (or both).

(6) θ is of the form $\phi \leftrightarrow \psi$. θ is true on $\mathfrak{m}$ iff ϕ and ψ have the same truth value on $\mathfrak{m}$.

(7) A sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$.

For every model $\mathfrak{m}$ for sentence ϕ this definition (i.e. the definition of truth) fixes a unique truth value for ϕ.[15]

Although there are an infinite number of SL sentences letters to which a model can assign truth values, the only ones that matter for any given sentence are, unsurprisingly, the sentence letters that the sentence contains.[16] For example, when assessing the value of $A \rightarrow B$ on $\mathfrak{m}$ only the assignments to A and B are relevant; assignments to other letters are irrelevant. It follows that if two models $\mathfrak{m}_1$ and $\mathfrak{m}_2$ assign the same truth values to A and B, respectively, then they fix the same truth value for $A \rightarrow B$.

[13] This is the definition of "model" in many logic textbooks. We prefer to include models that make assignments to only a subset of sentence letters.

[14] We discuss *truth functions* further in section 2.2.2.

[15] If a model $\mathfrak{m}$ is *not* a model for some SL sentence ϕ then $\mathfrak{m}$ does *not* fix a truth value for ϕ.

[16] We prove this later in the chapter.

If $\mathfrak{m}$ makes assignments to A and B but no other sentence letters, we say that $\mathfrak{m}$ is a *minimal model* of $A \to B$. More generally:

Definition 2.22. $\mathfrak{m}$ is a *minimal model of* ϕ iff $\mathfrak{m}$ makes assignments to every sentence letter in ϕ but to no other sentence letters.

There are only four minimal models for the sentence $A \to B$, because there are only 4 combinations of truth values that can be assigned to A and B. The number of distinct minimal models for a sentence ϕ is 2^n, where n is the number of sentence letters (counted by type) in ϕ.

There are several ways to compute the truth value of an SL sentence in a model. We demonstrate some informal ones in the following examples, and then develop a systematic method in section 2.2.4 (on page 28).

Example 2.23. Give the truth value of $A \vee \sim B$ on a model $\mathfrak{m}$ such that $\mathfrak{m}(A) = \mathsf{F}$ and $\mathfrak{m}(B) = \mathsf{F}$.

Proof: Given that $\mathfrak{m}(B) = \mathsf{F}$, it follows by the negation clause of the definition of truth that $\sim B$ is true on $\mathfrak{m}$. And since $\sim B$ is true on $\mathfrak{m}$, it follows by the disjunction clause of the definition of truth that $A \vee \sim B$ is too. ■

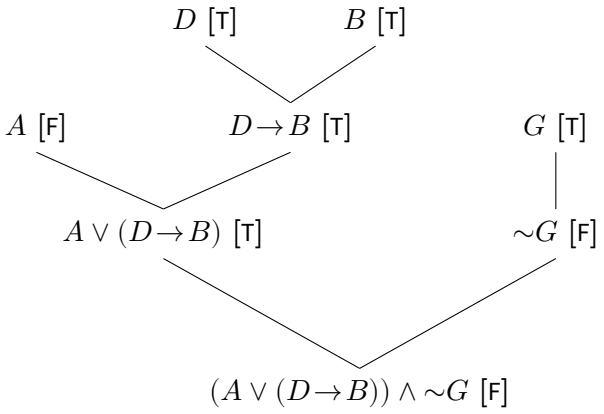
> One way to compute the truth value of this sentence in $\mathfrak{m}$ is to read off the values of the atomic subsentences and use definition 2.21 (the definition of truth in SL) to determine the value of successively larger subsentences, until finally you get the value of the whole sentence.

Example 2.24. Give the truth value of $(A \vee (D \to B)) \wedge \sim G$ on a model $\mathfrak{m}$ such that $\mathfrak{m}(A) = \mathsf{F}$, $\mathfrak{m}(D) = \mathsf{T}$, $\mathfrak{m}(B) = \mathsf{T}$, and $\mathfrak{m}(G) = \mathsf{T}$.

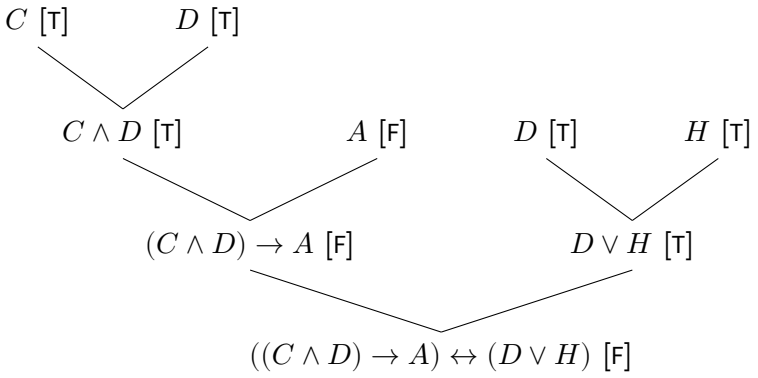
Proof: Since $\mathfrak{m}(G) = \mathsf{T}$, it follows by the negation clause of the definition of truth that $\sim G$ is false on $\mathfrak{m}$. So, by the conjunction clause of the definition of truth, $(A \vee (D \to B)) \wedge \sim G$ is false too. ■

> When computing the truth value for a complicated sentence it can help to be strategic about which subsentences to look at first. Often you don't need to determine the value of every subsentence. The main connective can be an important clue about where to start. The sentence $(A \vee (D \to B)) \wedge \sim G$ is a conjunction, which is false on a model if any one of its conjuncts is.

We can use a construction tree to make sure we compute the truth values of the subsentences in an appropriate order. The idea is to start at the top of the construction tree, the truth values of which are given by the model, and work our way down the branches. Let's illustrate with $(A \vee (D \to B)) \wedge \sim G$ on $\mathfrak{m}$:



Example 2.25. Compute the truth value of the sentence $((C \wedge D) \to A) \leftrightarrow (D \vee H)$ on a model $\mathfrak{m}$ such that $\mathfrak{m}(C) = \mathsf{T}$, $\mathfrak{m}(D) = \mathsf{T}$, $\mathfrak{m}(A) = \mathsf{F}$, and $\mathfrak{m}(H) = \mathsf{T}$.



2.2.2 Truth Functions and Truth Tables

A truth function is any function $f : \{\mathsf{T}, \mathsf{F}\} \times ... \times \{\mathsf{T}, \mathsf{F}\} \Rightarrow \{\mathsf{T}, \mathsf{F}\}$, i.e., from sequences of truth values to truth values. The definition of truth in a model (2.21) associates each logical connective of SL with a truth function. For example, the truth function for $\to$ is:

$$\begin{aligned} f(\mathsf{T}, \mathsf{T}) &= \mathsf{T} \\ f(\mathsf{T}, \mathsf{F}) &= \mathsf{F} \\ f(\mathsf{F}, \mathsf{T}) &= \mathsf{T} \\ f(\mathsf{F}, \mathsf{F}) &= \mathsf{T} \end{aligned}$$

Or, for short:

$$f(v_1, v_2) = \begin{cases} \mathsf{F} & \text{if } v_1 = \mathsf{T} \text{ and } v_2 = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$$

A truth function can also be given as a *truth table*. The truth table for $\rightarrow$ is:

$\rightarrow$	T	F
T	T	F
F	T	T

or can be written alternatively as:

ϕ	ψ	$\phi \rightarrow \psi$
T	T	T
T	F	F
F	T	T
F	F	T

2.2.3 Logical Truth: TFT, TFF, & TFC

A randomly chosen sentence will probably be true on some models and false on others. However, some sentences are true on all models. One example of such is the sentence $A \vee \sim A$. Others are false on all models, e.g. $A \wedge \sim A$.

Definition 2.26. A sentence ϕ of SL is *truth functionally true* (TFT) iff it is true on all models for ϕ.

Example 2.27. Prove that $A \vee \sim A$ is TFT.

Proof: A model $\mathfrak{m}$ for $A \vee \sim A$ has to assign either T or F to A. If it assigns T to A, then $A \vee \sim A$ is true on $\mathfrak{m}$. Otherwise it assigns F to A, in which case $\sim A$ is true on $\mathfrak{m}$. It follows that $A \vee \sim A$ is true on $\mathfrak{m}$. Either way, the sentence is true on $\mathfrak{m}$. This holds in all models $\mathfrak{m}$, so $A \vee \sim A$ is TFT. ■

Example 2.28. Prove that $B \rightarrow (C \rightarrow B)$ is TFT.

Proof: Any model $\mathfrak{m}$ will assign either T or F to B. If it assigns F to B, then $B \rightarrow (C \rightarrow B)$ is true on $\mathfrak{m}$, because, according to the def. of truth for $\rightarrow$, a conditional is true if the LHS is false. If $\mathfrak{m}$ assigns T to B, then $C \rightarrow B$ is true on $\mathfrak{m}$, again because of the def. of truth for $\rightarrow$. But if $C \rightarrow B$ is true on $\mathfrak{m}$, it follows that $B \rightarrow (C \rightarrow B)$ is true on $\mathfrak{m}$. ■

Definition 2.29. A sentence ϕ of SL is *truth functionally false* (TFF) iff it is false on all models for ϕ.

Example 2.30. Prove that $A \wedge \sim A$ is TFF.

Proof: A model $\mathfrak{m}$ for $A \wedge \sim A$ has to assign either T or F to A. If it assigns T to A, then $\sim A$ is false on $\mathfrak{m}$. So $A \wedge \sim A$ is false on $\mathfrak{m}$. But if $\mathfrak{m}$ assigns F to A, then $A \wedge \sim A$

is false on $\mathfrak{m}$. Either way, the sentence is false on $\mathfrak{m}$. This holds in all models $\mathfrak{m}$, so $A \wedge \sim A$ is TFF. ■

Example 2.31. Prove that $\sim C \wedge ((B \rightarrow C) \wedge B)$ is TFF.

Proof: Assume that the sentence *isn't* TFF. Then there is some model $\mathfrak{m}$ that makes it true. By def. of truth for $\wedge$, both $\sim C$ and $(B \rightarrow C) \wedge B$ are true on $\mathfrak{m}$. Since $\sim C$ is true on $\mathfrak{m}$, C is false on $\mathfrak{m}$. Since $(B \rightarrow C) \wedge B$ is true on $\mathfrak{m}$, then both $B \rightarrow C$ and B are true on $\mathfrak{m}$. Thus C is true on $\mathfrak{m}$ too. But $\mathfrak{m}$ can't assign both `F` and `T` to C. So there cannot be any model $\mathfrak{m}$ that makes $\sim C \wedge ((B \rightarrow C) \wedge B)$ true. Therefore it's TFF. ■

> This is an *indirect proof*. To use an indirect proof, start by assuming the opposite of what you want to prove. Then show how that assumption leads to a contradiction. No contradiction can be true. So, if an assumption leads to a contradiction then it must be false. The opposite of the assumption must therefore be true.
>
> Another name for indirect proof is '*reductio ad absurdum*' (sometimes just '*reductio*' or 'RAA'). For many problems and theorems, RAA is the easiest method to use. It is called '*reductio ad absurdum*' because it 'reduces' the initial assumption to a contradiction, an absurdity.

Definition 2.32. A sentence ϕ of SL is *truth functionally contingent* (TFC) iff it is true on one model for ϕ and false on another.

Example 2.33. In example 2.23 (on page 24) we saw that the sentence $(A \vee (D \rightarrow B)) \wedge \sim G$ is false on one model. To see that it can also be true, and thus that the sentence is TFC, consider the following model: $\mathfrak{m}(A) = \mathtt{F}$, $\mathfrak{m}(D) = \mathtt{T}$, $\mathfrak{m}(B) = \mathtt{T}$, and $\mathfrak{m}(G) = \mathtt{T}$. Convince yourself that $\mathfrak{m}$ makes $(A \vee (D \rightarrow B)) \wedge \sim G$ true.

Every sentence of SL is either TFT, TFF, or TFC.

Although the definitions of TFT, TFF, and TFC are specific to SL—the models to which each definition refers are models of SL—we can use essentially the same definitions for *any* formal language, as long as there is some notion of a model for that language. We can think of sentences which fit these definitions as being (respectively) *logically true*, *logically false*, and *logically contingent*. Hereafter we sometimes use the more general term 'logical truth' instead of 'truth functional truth'.[17]

[17] A logical truth is sometimes said to be *valid*, or said to be a *tautology*. We avoid using these terms for logical truths.

2.2.4 Procedures for Testing TFT, TFF, & TFC

In the above examples (2.27–2.31) we used the definition for truth (definition 2.21 (on page 23)) to show whether a given SL sentence was TFT, TFF, or TFC. But there are more systematic methods for classifying SL sentences.

Perhaps the most well known method involves using truth tables.[18] Later in this chapter we prove theorem 2.70 (on page 42), which says that the truth of a given SL sentence ϕ can be determined by a *minimal model*. Remember that a minimal model for ϕ assigns truth values only to the sentence letters appearing in ϕ. One way to test whether ϕ is TFT, TFF, or TFC is to write down all the possible assignments of truth values to the sentence letters of ϕ, then compute the truth value of ϕ for each assignment. The number of possible assignments (minimal models) is finite. For a sentence ϕ with n sentence letters (counted by type) there are 2^n possible assignments. If ϕ is true in all assignments, then ϕ is TFT. If it's false in all of them, then ϕ is TFF. And if it is true in some assignments and false in others, then ϕ is TFC.

Consider the sentence $(A \vee (D \rightarrow B)) \wedge \sim G$. Because this sentence has 4 sentence letters its table has $2^4 = 16$ rows. First write the 4 sentence letters on a top row. Then fill out the assignments by starting at the far right sentence letter (G) and putting alternating T's and F's below it until all 16 rows are filled. (See table 2.1.) Next,

A	B	D	G	$((A \vee (D \rightarrow B)) \wedge \sim G)$
T	T	T	T	F
T	T	T	F	T
T	T	F	T	F
T	T	F	F	T
T	F	T	T	F
T	F	T	F	T
T	F	F	T	F
T	F	F	F	T
F	T	T	T	F
F	T	T	F	T
F	T	F	T	F
F	T	F	F	T
F	F	T	T	F
F	F	T	F	F
F	F	F	T	F
F	F	F	F	T

Table 2.1: Sample Truth Table

move to the second-to-the-right sentence letter (D) and alternate by 2 T's and 2 F's

[18] Peirce (1902) was the first to use truth tables. See Hodges (2001b: 5).

for 16 rows. Move one more sentence letter to the left (B) and alternate T's and F's 4 at a time. Finally, alternate by 8 at a time for the left-most sentence letter (A). This completes the 16 rows so that each row is a unique assignment of truth values to the sentence letters. All 16 possible assignments are guaranteed to to be present. In general the pattern is to start at the far right column alternating T and F, then move to the left doubling the number of T's and F's that appeared in the previous column.

Then we write the sentence to the right of the sentence letters and under it put, in the respective rows, its truth value for each assignment. Once the truth value of the sentence is computed for all rows it's a trivial matter to determine from the table whether the sentence is TFT, TFF, or TFC. If T is under the sentence in every row, then it's TFT. If F is in every row, then it's TFF. And if each of T and F appear in at least one row, then it's TFC. Table 2.1 shows that $((A \vee (D \rightarrow B)) \wedge \sim G)$ is TFC.

The process of filling out a truth table is entirely 'mechanical.' We can carry out the process by following purely formal rules. Once we have a truth table for some SL sentence, we can again use purely formal rules to determine whether it's TFT, TFF, or TFC. No creativity is necessary to determine whether any given sentence is, e.g., TFT.

Example 2.34. We saw in example 2.27 (on page 26) that $A \wedge \sim A$ is TFF. The following truth table confirms this.

A	$A \wedge \sim A$
T	F
F	F

Example 2.35. In example 2.31 (on page 27) we saw that $B \rightarrow (C \rightarrow B)$ is TFT. This is confirmed by the following truth table.

B	C	$B \rightarrow (C \rightarrow B)$
T	T	T
T	F	T
F	T	T
F	F	T

Example 2.36. The sentence $\sim C \wedge ((B \rightarrow C) \wedge B)$ is TFF. The following truth table proves it.

B	C	$\sim C \wedge ((B \rightarrow C) \wedge B)$
T	T	F
T	F	F
F	T	F
F	F	F

While truth tables are a convenient method for answering many questions about SL sentences, two warnings are in order. First, for complex sentences the size of the truth table can be very large. The number of rows needed for a truth table grows exponentially with the number of sentence letters. Second, while truth tables are useful in SL, there is nothing comparable for the more sophisticated formal languages we cover in later chapters. The sooner you learn to analyze sentences directly, rather than relying on a truth table, the better. For example, consider the sentence $(E \rightarrow (B \wedge \sim(C \vee D))) \vee (A \rightarrow A)$. You *could* evaluate this sentence with a 32 line truth table. But it's much easier to prove that $(A \rightarrow A)$ is TFT, and then to show that it follows that the whole sentence is TFT.

The various formal derivation systems used in logic provide another class of procedures for testing SL sentences for TFT and TFF. In chapter 6 we develop one such system.[19] We also develop an algorithm in chapter 8 that can be used with our derivation system as a testing procedure for TFC sentences.

2.3 Entailment and other Relations

2.3.1 Entailment

Now we turn to the notion of entailment. We write that a set of sentences Δ entails a single sentence θ as follows: '$\Delta \models \theta$'. The symbol '$\models$', called the double turnstile, is *not* a symbol of SL. Like the Greek letters it is a symbol of MathEnglish.

Definition 2.37. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

(1) $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.

(2) $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Each of the following is a consequence of the definition of entailment for the case in which Δ is of finite size n:

(1) $\phi_1, \phi_2, \phi_3, \ldots, \phi_n \models \theta$ iff every model for ϕ_1, ϕ_2, ϕ_3, ..., ϕ_n, and θ that makes all of ϕ_1, ϕ_2, ϕ_3, ... ϕ_n true also makes θ true.

(2) $\phi_1, \phi_2, \phi_3, \ldots, \phi_n \models \theta$ iff every model for ϕ_1, ϕ_2, ϕ_3, ..., ϕ_n, and θ either makes at least one of ϕ_1, ϕ_2, ϕ_3, ... ϕ_n false or makes θ true.

[19] We use a natural deduction system. Natural deduction systems do not provide a procedure for testing whether sentences are TFC. Some derivation systems—e.g. semantic tableaux systems—do. A comprehensive introductory treatment of semantic tableaux (called truth trees by the author) is given in Nicholas J.J. Smith's (2012) textbook *Logic: The Laws of Truth*. Smith also gives a more detailed and thorough introduction to truth tables. Other derivation systems, such as Hilbert-style axiomatic systems, and Gentzen-style sequent calculi, don't on their own provide direct means of testing for TFC sentences.

Each of the following is another consequence, for the case in which Δ contains just one sentence:

(1) A sentence ϕ entails another sentence θ iff every model for ϕ and θ that makes ϕ true also makes θ true.

(2) A sentence ϕ entails another sentence θ iff there is no model for ϕ and θ that makes ϕ true and θ false.

Finally, Δ can be the empty set or an infinite set. When Δ is empty and $\Delta \models \theta$ we can write: $\models \theta$.

Theorem 2.38. For all SL sentences θ, $\models \theta$ iff θ is TFT.

Proof: ($\Rightarrow$) Assume that $\models \theta$. Then, by the definition of $\models$, every model for θ either makes a sentence to the left of the turnstile false or makes θ true. But there are no sentences to the left of the turnstile. So, every model for θ makes θ true. Thus θ is TFT.

($\Leftarrow$) Assume that θ is TFT. Then there is no model that makes θ false. Let Δ be the empty set. Then there is no model that makes all sentences in Δ true and θ false. So, by the definition of $\models$, $\Delta \models \theta$. And since Δ is empty, $\models \theta$.

Therefore $\models \theta$ iff θ is TFT. ■

> The statement of this theorem is a biconditional, i.e. a sentence of the form A iff B. The typical method for proving biconditionals is to divide the proof into two parts. In the first part, indicated by '($\Rightarrow$)', assume A and show that B follows. In the second part, indicated by '($\Leftarrow$)', assume B and show that A follows. After both parts are demonstrated, then conclude that A iff B.

Example 2.39. Prove $(A \wedge B) \models B$.

Proof: Let $\mathfrak{m}$ be a model that makes $(A \wedge B)$ true. By the definition of truth for $\wedge$—clause 3, 2.21 (on page 23)—both A and B are true in $\mathfrak{m}$ as well. Thus any model that makes $(A \wedge B)$ true also makes B true. Therefore, by the definition of $\models$, $(A \wedge B) \models B$. ■

> One method for proving an entailment is to assume a model that makes all sentences to the left of the turnstile true, and then to show that the sentence on the right must therefore also be true.

Example 2.40. Show that $B \models (A \vee B)$. We leave this as an exercise the reader.

Example 2.41. Show that for all ϕ there is some θ such that $\phi \models \theta$. That is, that every SL sentence entails at least one SL sentence.

Proof: Let ϕ be any SL sentence. Then let $\theta = \phi$. Then there is no model that makes ϕ true and θ false, since they are the same sentence. Therefore, by the definition of $\models$, $\phi \models \theta$. Nothing particular was assumed about ϕ, so the argument holds for all SL sentences. ■

> We must show that no matter what SL sentence ϕ you pick, there's always some SL sentence θ entailed by it. But this is easy: every SL sentence entails itself. Thus, no matter what SL sentence you pick there's always at least one sentence entailed by it.

Example 2.42. Show that there is some θ such that for all ϕ, $\phi \models \theta$. That is, show there's some SL sentence that's entailed by all SL sentences.

Proof: Let θ be $(A \vee \sim A)$. It was shown previously that $(A \vee \sim A)$ is TFT. Then, by the definition of TFT, θ is true on all models. So there is no model such that ϕ is true and θ is false. Thus, by the definition of $\models$, $\phi \models \theta$. This argument holds for all ϕ. ■

Example 2.43. Show that there is some ϕ such that, for all θ, $\phi \models \theta$. We leave the proof as an exercise for the reader.

2.3.2 Procedures for Testing Entailment

In the previous examples (2.39–2.43) we used the definition of entailment to prove entailment claims. But there are other methods for checking whether a given entailment holds. Truth tables provide a simple, mechanical procedure.

To test whether a set of sentences Δ entails a sentence θ, we first write down all the sentence letters of θ and Δ. We put these on the left side of the truth table and under them (following the procedure outlined in section 2.2.4, on page 28) we write all the possible assignments of truth values. Then, to the right of the sentence letters, we write each of the sentences from Δ (in separate columns), and to the right of those we write θ. Under θ and each of the sentences from Δ we write the truth value of that sentence based on the assignment of truth values listed in that row. Δ entails θ iff there is no row in the truth table in which all the sentences in Δ are true and θ is false. Any rows of the truth table that do not make all of the LHS sentences true are irrelevant. After marking one of the LHS sentences as false we can omit the rest of the row.

Example 2.44. In example 2.39 we saw that $(A \wedge B) \models B$. We show this with a truth table.

A	B	$(A \wedge B)$	B
T	T	T	T
T	F	F	F
F	T	F	T
F	F	F	F

Example 2.45. In example 2.40 we said that $B \models (A \vee B)$. The following truth table shows this.

A	B	B	$(A \vee B)$
T	T	T	T
T	F	F	T
F	T	T	T
F	F	F	F

Example 2.46. We conclude with a more complicated example. Here we show that $A \vee B, {\sim}C \rightarrow {\sim}A, B \rightarrow C \models C$.

A	B	C	$A \vee B$	${\sim}C \rightarrow {\sim}A$	$B \rightarrow C$	C
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	T	T	T	T
T	F	F	T	F	T	F
F	T	T	T	T	T	T
F	T	F	T	T	F	F
F	F	T	F	T	T	T
F	F	F	F	T	T	F

2.3.3 Basic Results on Entailment

A simple, but important theorem involves moving sentences from one side of the turnstile to the other.

Theorem 2.47. SL Exportation Theorem: For all SL sentences ϕ and θ, $\phi \models \theta$ iff $\models \phi \rightarrow \theta$.

> The statement of this theorem is a biconditional, so we prove it as we proved the last biconditional theorem. There are two parts. First, we assume that the left-hand side, $\phi \models \theta$, is true, and show that $\models \phi \rightarrow \theta$ follows from it. Second, we assume that $\models \phi \rightarrow \theta$ is true, and show that $\phi \models \theta$ follows. That is sufficient to demonstrate that $\phi \models \theta$ iff $\models \phi \rightarrow \theta$.

Proof: ($\Rightarrow$) Assume that $\phi \models \theta$. Then, by the definition of $\models$ it follows that on every model on which ϕ is true, θ is also true. According to the definition of truth for $\rightarrow$, $\mathfrak{m}$ makes $(\phi \rightarrow \theta)$ false only if it makes ϕ true and θ false. But we have already shown that there is no such model. It follows that $(\phi \rightarrow \theta)$ is TFT (def. of TFT, 2.26, on page 26). And according to theorem 2.38 (on page 31), it follows that the entailment $\models (\phi \rightarrow \theta)$ holds.

($\Leftarrow$) Assume that $\models(\phi\rightarrow\theta)$. By the definition of $\models$, since $\models(\phi\rightarrow\theta)$ it follows that $(\phi\rightarrow\theta)$ is TFT. So, by the definition of TFT, $(\phi\rightarrow\theta)$ is true on every model $\mathfrak{m}$. By the definition of truth of $\rightarrow$, it follows that there is no model that makes ϕ true and θ false. So, by the definition of $\models$, that means that $\phi\models\theta$. ■

There are a number of other theorems that are similar to, and expand upon, theorem 2.47. Here we state them and leave the proofs to the reader.

Theorem 2.48.

(1) If ϕ_1, ϕ_2, ..., ϕ_n and ψ are SL sentences, then

$$\begin{aligned}
&\phi_1,\phi_2,\dots,\phi_n\models\psi \text{ iff } \phi_2,\dots,\phi_n\models(\phi_1\rightarrow\psi)\\
&\phi_1,\phi_2,\dots,\phi_n\models\psi \text{ iff } \phi_1,\phi_3,\dots,\phi_n\models(\phi_2\rightarrow\psi)\\
&\qquad\qquad\vdots\\
&\phi_1,\phi_2,\dots,\phi_n\models\psi \text{ iff } \phi_1,\dots,\phi_{n-1}\models(\phi_n\rightarrow\psi)
\end{aligned}$$

(2) If ϕ_1, ϕ_2, ..., ϕ_n and ψ are SL sentences, then

$$\begin{aligned}
&\phi_1,\phi_2,\dots,\phi_n\models\psi \text{ iff } \phi_1,\dots,\phi_{n-1}\models(\phi_n\rightarrow\psi)\\
&\phi_1,\phi_2,\dots,\phi_n\models\psi \text{ iff } \phi_1,\dots,\phi_{n-2}\models(\phi_{n-1}\rightarrow(\phi_n\rightarrow\psi))\\
&\qquad\qquad\vdots\\
&\phi_1,\phi_2,\dots,\phi_n\models\psi \text{ iff } \models(\phi_1\rightarrow(\dots\rightarrow(\phi_{n-1}\rightarrow(\phi_n\rightarrow\psi))))
\end{aligned}$$

(3) If ϕ and ψ are SL sentences and Δ is a set of SL sentences containing ϕ (i.e, $\phi\in\Delta$) and Δ^* is Δ with ϕ removed, then $\Delta\models\psi$ iff $\Delta^*\models(\phi\rightarrow\psi)$.

(4) If ϕ and ψ are SL sentences, then $\phi\models\psi$ iff $\phi,\sim\psi\models(A\wedge\sim A)$.

(5) If $\phi_1,\dots,\phi_n$ and $\psi_1,\dots,\psi_n$ are all SL sentences, then

$$\begin{aligned}
&\phi_1,\dots,\phi_n\models(\psi_1\vee\dots\vee\psi_n) \text{ iff } \phi_1,\dots,\phi_n,\sim\psi_1\models(\psi_2\vee\dots\vee\psi_n)\\
&\phi_1,\dots,\phi_n\models(\psi_1\vee\dots\vee\psi_n) \text{ iff } \phi_1,\dots,\phi_n,\sim\psi_2\models(\psi_1\vee\psi_3\vee\dots\vee\psi_n)\\
&\qquad\qquad\vdots\\
&\phi_1,\dots,\phi_n\models(\psi_1\vee\dots\vee\psi_n) \text{ iff } \phi_1,\dots,\phi_n,\sim\psi_n\models(\psi_1\vee\dots\vee\psi_{n-1})
\end{aligned}$$

2.3.4 Other Relations

There are a number of other important relations between SL sentences:

Definition 2.49. Two sentences θ and ϕ are *truth functionally equivalent* (TFE) iff for all models $\mathfrak{m}$ for θ and ϕ, θ and ϕ have the same truth value on $\mathfrak{m}$.

Theorem 2.50. θ and ϕ are TFE iff $\theta\models\phi$ and $\phi\models\theta$.

Proof: ($\Rightarrow$) Assume that θ and ϕ are TFE. Then, by the definition of TFE, there is no model on which θ and ϕ have different truth values. So there is no model on which θ is true and ϕ is false. Then, by the definition of $\models$, $\theta \models \phi$. And there is no model on which ϕ is true and θ is false. Then, by the definition of $\models$, $\phi \models \theta$.

($\Leftarrow$) Assume that $\theta \models \phi$ and $\phi \models \theta$. Since $\theta \models \phi$, then by the definition of $\models$ every model that makes θ true also makes ϕ true. And since $\phi \models \theta$, then by the definition of $\models$ every model that makes ϕ true also makes θ true. Since each sentence is true when the other is, and a sentence is false iff it's not true, θ and ϕ must be the same value on all models. So they are TFE ■

Definition 2.51. Two sentences θ and ϕ are *truth functionally contradictory* iff for all models $\mathfrak{m}$ for θ and ϕ, θ and ϕ have opposite truth values on $\mathfrak{m}$.

Theorem 2.52. θ and ϕ are truth functionally contradictory iff θ is TFE to $\sim\phi$.

Proof: ($\Rightarrow$) Assume that θ and ϕ are truth functionally contradictory. Then θ and ϕ have opposite truth values on all models. By the definition of truth for $\sim$, ϕ and $\sim\phi$ have opposite truth values on all models. So on a model that makes θ true, ϕ is false and $\sim\phi$ is true. And on a model that makes θ false, ϕ is true and $\sim\phi$ is false. It follows that θ and $\sim\phi$ have the same value on all models. Thus, by the definition of TFE, θ is TFE to $\sim\phi$.

($\Leftarrow$) Assume that θ is TFE to $\sim\phi$. Then by the definition of TFE θ and $\sim\phi$ have the same truth value on all models. By the definition of truth for $\sim$, ϕ and $\sim\phi$ have opposite truth values on all models. So on a model that makes θ true, ϕ is true and $\sim\phi$ is false. And on a model that makes θ false, ϕ is false and $\sim\phi$ is true. It follows that θ and ϕ have opposite values on all models. Thus, θ is truth functionally contradictory to ϕ.

■

Definition 2.53. Two sentences θ and ϕ are *truth functionally contrary* iff there is no model on which both are true.

Example 2.54. θ and ϕ are truth functionally contrary iff $\theta, \phi \models A \wedge \sim A$.

Proof: ($\Rightarrow$) Assume that θ and ϕ are truth functionally contrary. Then there is no model on which θ and ϕ are both truth. It follows that there is no model that makes both θ and ϕ true and $A \wedge \sim A$ false. Thus, by the definition of $\models$, $\theta, \phi \models A \wedge \sim A$.

($\Leftarrow$) This direction is left as an exercise for the reader. ■

Definition 2.55. Two sentences θ and ϕ are *truth functionally subcontrary* iff there is no model on which both are false.

Theorem 2.56. θ and ϕ are truth functionally subcontrary iff $\models \theta \vee \phi$. This proof is left as an exercise for the reader.

Definition 2.57. Two sentences θ and ϕ are *truth functionally independent* iff there are four models:

(1) A model in which both θ and ϕ are true;

(2) A model in which both θ and ϕ are false;

(3) A model in which θ is true and ϕ is false; and

(4) A model in which θ is false and ϕ is true.

It follows truth functionally independent sentences are not truth functionally equivalent, contradictory, contrary, or subcontrary.

Example 2.58. Each pair of contradictory sentences is also contrary, but sentences can be contrary without being contradictory.

Proof: $C \wedge D$ and $C \wedge \sim D$ are contrary but not contradictory.

(Contrary:) If $C \wedge D$ is true in a model $\mathfrak{m}$, then C and D are each true on $\mathfrak{m}$. So $\sim D$ is false in $\mathfrak{m}$, and $C \wedge \sim D$ is false in $\mathfrak{m}$. If $C \wedge \sim D$ is true in $\mathfrak{m}$, then both C and $\sim D$ are true on $\mathfrak{m}$. D is false in $\mathfrak{m}$, and hence $C \wedge D$ is false in $\mathfrak{m}$. Thus, by def. 2.53, the pair is contrary.

(Not Contradictory:) Any model $\mathfrak{m}$ that assigns `F` to C makes both $C \wedge D$ and $C \wedge \sim D$ false. By def. 2.51, the pair is not contradictory. ■

Example 2.59. Contradictory sentences are also subcontrary, but sentences can be subcontrary without being contradictory.

Proof: D and $C \vee \sim D$ are subcontrary but not contradictory.

(Subcontrary:) Assume that D is false on a model $\mathfrak{m}$. It follows that $\sim D$ is true on $\mathfrak{m}$ and so is $C \vee \sim D$. Alternatively, assume that $C \vee \sim D$ is false on $\mathfrak{m}$. Then both C and $\sim D$ are false on $\mathfrak{m}$. So D is true on $\mathfrak{m}$. By def. 2.55, the pair is subcontrary.

(Not Contradictory:) Any model that assigns `T` to both D and C makes both D and $C \vee \sim D$ true. So by def. 2.51, the pair isn't contradictory. ■

Example 2.60. If two sentences are both contrary and subcontrary, they are contradictory.

Proof: If two sentences are contrary, then by definition 2.53 any model $\mathfrak{m}$ that makes one true makes the other false. If two sentences are subcontrary, then by definition 2.55 any model $\mathfrak{m}$ that makes one false makes the other true. Because every model either makes a sentence true or makes it false, no model assigns two sentences that are contrary and subcontrary the same truth value. So by definition 2.51, two sentences that are contrary and subcontrary are also contradictory. ■

Example 2.61. $A \vee (B \wedge D)$ and $(A \vee B) \wedge (A \vee D)$ are TFE.

Proof: ($\Rightarrow$) Assume that $A \vee (B \wedge D)$ is true on some $\mathfrak{m}$. By the def. of truth of $\vee$, either A is true on $\mathfrak{m}$ or $B \wedge D$ is true on $\mathfrak{m}$. (Case 1) A is true on $\mathfrak{m}$. Then both $A \vee B$ and $A \vee D$ are true on $\mathfrak{m}$. So, $(A \vee B) \wedge (A \vee D)$ is true on $\mathfrak{m}$. (Case 2) $B \wedge D$ is true on $\mathfrak{m}$ Then then both B and D are true on $\mathfrak{m}$, and so both $A \vee B$ and $A \vee D$ are true on $\mathfrak{m}$. Hence, $(A \vee B) \wedge (A \vee D)$ is true on $\mathfrak{m}$.

In either case, $(A \vee B) \wedge (A \vee D)$ is true on $\mathfrak{m}$. Thus, $A \vee (B \wedge D) \models (A \vee B) \wedge (A \vee D)$.

($\Leftarrow$) We leave it to the reader to show that $(A \vee B) \wedge (A \vee D) \models A \vee (B \wedge D)$. It then follows by theorem 2.50 that the two sentences are TFE. ■

Example 2.62. For any SL sentences ϕ and θ, $\phi \rightarrow \theta$ and $\sim\phi \vee \theta$ are TFE.

Proof: ($\Rightarrow$) Assume that $\phi \rightarrow \theta$ is true on some model $\mathfrak{m}$. Then on $\mathfrak{m}$ either ϕ is false or θ is true. Hence either $\sim\phi$ or θ is true on $\mathfrak{m}$. It follows that $\sim\phi \vee \theta$ is true on $\mathfrak{m}$. Hence by the definition of $\models$, $\phi \rightarrow \theta \models \sim\phi \vee \theta$.

($\Leftarrow$) We leave it to the reader to show that $\sim\phi \vee \theta \models \phi \rightarrow \theta$. It then follows by theorem 2.50 that the two sentences are TFE. ■

Example 2.63. For any SL sentence ϕ, $\sim\sim\phi$ and ϕ are TFE.

Proof: The proof is left to the reader. ■

2.3.5 Procedures for Testing Other Relations

As before we can use truth tables to test for the following relationships: truth-functional equivalence, truth-functional contradictory, truth-functional contrary, truth-functional subcontrary, and truth-functional independence. We can test for truth-functional equivalence by putting both ϕ and θ in a truth table and checking whether they have the same truth value in every row.

Example 2.64. In example 2.61 (on the previous page) we saw that $A \vee (B \wedge D)$ and $(A \vee B) \wedge (A \vee D)$ are TFE. We can also prove this with a truth table.

A	B	D	$A \vee (B \wedge D)$	$(A \vee B) \wedge (A \vee D)$
T	T	T	T	T
T	T	F	T	T
T	F	T	T	T
T	F	F	T	T
F	T	T	T	T
F	T	F	F	F
F	F	T	F	F
F	F	F	F	F

Example 2.65. In example 2.62 (on this page) we saw that for any SL sentences ϕ and θ, $\phi \rightarrow \theta$ and $\sim\phi \vee \theta$ are TFE. We confirm with a truth table:

ϕ	θ	$\phi \rightarrow \theta$	$\sim\phi \vee \theta$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

In this example we're not looking at SL sentences; instead we have sentence schemas. But for reasons to be discussed below this procedure still works: this truth table shows that for any two sentences ϕ and θ, $\phi \rightarrow \theta$ and $\sim\phi \vee \theta$ are TFE.

While truth tables provide guaranteed answers to these questions, students should not become reliant on this method. For one thing, this method doesn't work for more complex languages, and it is important to practice with SL structures the reasoning skills and methods needed in later chapters. Secondly, truth tables can quickly become very large and tedious, and often simple reasoning suffices. More generally, truth tables give an answer but often don't give insight.

2.4 Recursive Proofs

2.4.1 The Method of Recursive Proof

Many logic concepts are characterized by recursive definitions. To prove a theorem about a recursively defined concept, we usually want to employ a method called *recursive proof*. The structure of a recursive proof mirrors the structure of a recursive definition.

Definition 2.66. Let Δ be some set whose members are defined recursively. A *recursive proof* that all members of Δ have some property ϕ proceeds as follows:

Base Step: Show that everything identified by the base clause of the recursive definition has ϕ.

Inheritance Step: Show that ϕ is inherited; i.e., show that if the previous elements from which new elements are generated (or found) by the generating clause have ϕ then the new ones have ϕ too.

Closure Step: Finally, show that the base and inheritance steps are sufficient to show that all elements of Δ have ϕ.

The inheritance step usually has two parts. The first part is a *recursive assumption*. We *assume* that some previous elements—the elements from which new ones are generated by the generating clause—already have the property in question. Typically we make this assumption by selecting metavariables to represent the previous elements. We are entitled to this assumption because we know that there are some previous elements. At the very least the elements identified in the base clause have

the property. Second, we prove that the new, generated elements must also have the property in question.

How much we need to write in the closure step varies from proof to proof. Sometimes we simply note that the closure condition ("that's all") ensures nothing has been left out. Other times, if the proof is more complicated, we might also reiterate what we have just shown.

2.4.2 Recursive Proof and Mathematical Induction

Many important mathematical concepts are defined recursively. We saw the recursive definition of natural numbers in the previous chapter. If we want to *prove* something about all natural numbers, we can use mathematical induction. *Mathematical induction* is a method of proof in which one shows (i) that some property holds of the first natural number, and (ii) that for any natural number n having that property, the successor of n also has that property. Mathematical induction is one sort of recursive proof:

Base Step: 0 has property ϕ.

Inheritance Step: Whenever n has property ϕ, its successor, $n+1$, also has ϕ.

Therefore, we conclude that

Closure Step: All natural numbers have property ϕ.

Let's go through an example of mathematical induction. This first proof isn't exciting but it illustrates how recursive proofs work.

Example 2.67. There is no largest natural number n.

Proof:

(1) Base Step: The number 0 isn't the largest; 1 is larger.

(2) Inheritance Step: Assume that n isn't the largest natural number. (This is the recursive assumption.) Is $n+1$ the largest natural number? No. $n+1$ can't be the largest because $n+1 < n+2$.

(3) Closure Step: Therefore no natural number n is the largest.

■

Two things can give people trouble with recursive proofs. One is that the base case is often easy or trivial. Make sure you have done it correctly, but don't worry if it seems too easy.

Second, some students initially think that the recursive assumption in the inheritance step assumes what we are trying to prove. But in a correct proof that isn't so. Think of the recursive assumption this way: *if* some elements have property ϕ, then we can show that other elements also have ϕ. We know that some elements have

the property in question: those identified in the base step. The recursive assumption merely provides a label for these previously identified elements bearing the property in question.

2.5 Recursive Proofs in SL

2.5.1 Recursive Proof Examples in SL

Let's use recursive proofs to establish two simple results about SL sentences.

Example 2.68. Prove that every (official) sentence has exactly as many left parentheses as right.

Proof:

Base Step: All atomic sentences have zero left parentheses and zero right parentheses. Clearly $0 = 0$.

Inheritance Step:

Recursive Assumption: Suppose ϕ and $\theta, \theta_1, \theta_2, \ldots, \theta_n$ are sentences of order k or less, each with exactly as many left parentheses as right. Sentences of order $k+1$ can be constructed as follows:

Negation: Adding a negation sign to ϕ does not change the number of parentheses. Since, by recursive assumption (RA), ϕ has the same number of left and right parentheses, so does $\sim\phi$.

Conditional/Biconditional: By RA, each of ϕ and θ has matching numbers of left and right parentheses. It follows that the number of left parentheses in both sentences matches the number of right parentheses in both. The sentences $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$ have an additional pair of parentheses, one left and one right, and this preserves the property of having the same number of left and right parentheses.

Disjunction/Conjunction: By RA, each of $\theta_1, \theta_2, \ldots, \theta_n$ has a matching number of left and right parentheses. It follows that the number of left parentheses in all of them matches the number of right parentheses in all of them. The sentences $(\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_n)$ and $(\theta_1 \vee \theta_2 \vee \ldots \vee \theta_n)$ have one additional pair of parentheses, one left and one right, the addition of which preserves the property that the number of left parentheses is the same as the number of right parentheses.

Closure Step: The above steps cover all the ways of generating an SL sentence. In each case the number of left parentheses equals the number of right parentheses.

■

This recursive proof is structured to match the definition of an SL sentence, which is typical of proofs that all SL sentences have some property. The definition of a sentence has a base clause for sentence letters, and generating clauses for all the ways to construct a sentence of order $k+1$ from subsentences that have order k or less. Accordingly, the recursive assumption is nearly always of the form "Sentences of order k or less have such and such property." The goal in the rest of the inheritance step is to show that all the ways of constructing a sentence of order $k+1$ preserve that property.

Example 2.69. Prove that in every official SL sentence the number of left parentheses is greater than or equal to the number of arrows.

Proof: Let $\text{LP}\phi$ be the number of left parentheses in ϕ and $\text{AR}\phi$ be the number of arrows in ϕ.

Base Step: In the base case ϕ is atomic, so $\text{LP}\phi = 0$ and $\text{AR}\phi = 0$. Thus, $\text{LP}\phi = \text{AR}\phi$ and so $\text{LP}\phi \geq \text{AR}\phi$.

Inheritance Step:

Recursive Assumption: Let θ, θ_1, θ_2, ... θ_n be sentences of order k or less. Assume that $\text{LP}\theta \geq \text{AR}\theta$, $\text{LP}\theta_1 \geq \text{AR}\theta_1$, $\text{LP}\theta_2 \geq \text{AR}\theta_2$, ... $\text{LP}\theta_n \geq \text{AR}\theta_n$. Sentences of order $k+1$ can be constructed as follows:

Negation: $\text{LP}{\sim}\theta = \text{LP}\theta$ and $\text{AR}{\sim}\theta = \text{AR}\theta$. By assumption $\text{LP}\theta \geq \text{AR}\theta$, so $\text{LP}{\sim}\theta \geq \text{AR}{\sim}\theta$ too.

Conditional: $\text{LP}(\theta_1 \to \theta_2) = \text{LP}\theta_1 + \text{LP}\theta_2 + 1$ and $\text{AR}(\theta_1 \to \theta_2) = \text{AR}\theta_1 + \text{AR}\theta_2 + 1$. Because $\text{LP}\theta_1 \geq \text{AR}\theta_1$ and $\text{LP}\theta_2 \geq \text{AR}\theta_2$, clearly $\text{LP}\theta_1 + \text{LP}\theta_2 + 1 \geq \text{AR}\theta_1 + \text{AR}\theta_2 + 1$ too. So $\text{LP}(\theta_1 \to \theta_2) \geq \text{AR}(\theta_1 \to \theta_2)$.

Biconditional: $\text{LP}(\theta_1 \leftrightarrow \theta_2) = \text{LP}\theta_1 + \text{LP}\theta_2 + 1$ and $\text{AR}(\theta_1 \leftrightarrow \theta_2) = \text{AR}\theta_1 + \text{AR}\theta_2$. Because $\text{LP}\theta_1 \geq \text{AR}\theta_1$ and $\text{LP}\theta_2 \geq \text{AR}\theta_2$, clearly $\text{LP}\theta_1 + \text{LP}\theta_2 + 1 \geq \text{AR}\theta_1 + \text{AR}\theta_2$ too. So $\text{LP}(\theta_1 \leftrightarrow \theta_2) \geq \text{AR}(\theta_1 \leftrightarrow \theta_2)$.

Disjunction: We have that $\text{LP}(\theta_1 \vee \theta_2 \vee ... \vee \theta_n) = \text{LP}\theta_1 + \text{LP}\theta_2 + ... + \text{LP}\theta_n + 1$ and $\text{AR}(\theta_1 \vee \theta_2 \vee ... \vee \theta_n) = \text{AR}\theta_1 + \text{AR}\theta_2 + ... + \text{AR}\theta_n$. Because $\text{LP}\theta_1 \geq \text{AR}\theta_1$, $\text{LP}\theta_2 \geq \text{AR}\theta_2$, ... $\text{LP}\theta_n \geq \text{AR}\theta_n$, we have that $\text{LP}\theta_1 + \text{LP}\theta_2 + ... + \text{LP}\theta_n + 1 \geq \text{AR}\theta_1 + \text{AR}\theta_2 + ... + \text{AR}\theta_n$. So, $\text{LP}(\theta_1 \vee \theta_2 \vee ... \vee \theta_n) \geq \text{AR}(\theta_1 \vee \theta_2 \vee ... \vee \theta_n)$ too.

Conjunction: The argument is the same as that for disjunction, but with each '$\vee$' replaced with '$\wedge$'.

Closure Step: These are all the ways of constructing SL sentences, and in every case the number of left parentheses is greater than or equal to the number of arrows.

■

2.5.2 Minimal Model Theorem

Earlier we made the following claim: to determine whether some sentence ϕ is true on a model, you only need to check the assignments to the sentence letters in ϕ. Now we prove it.

Theorem 2.70. If two models for ϕ, $\mathfrak{m}_1$ and $\mathfrak{m}_2$, make the same assignments to all the sentence letters contained in ϕ, then ϕ is true on $\mathfrak{m}_1$ iff ϕ is true on $\mathfrak{m}_2$.

Proof:

Base Step: Let θ be a SL sentence of order 1. It follows that θ must be a lone sentence letter. If $\mathfrak{m}_1$ and $\mathfrak{m}_2$ make the same assignments for all the sentence letters, then θ, which is just one sentence letter, is true on $\mathfrak{m}_1$ iff θ is true on $\mathfrak{m}_2$.

Inheritance Step:

Recursive Assumption: Let ϕ and $\theta_1, \theta_2, \ldots, \theta_n$ be sentences of order k or less. Assume that ϕ is true on $\mathfrak{m}_1$ iff ϕ is true on $\mathfrak{m}_2$, θ_1 is true on $\mathfrak{m}_1$ iff θ_1 is true on $\mathfrak{m}_2$, ..., θ_n is true on $\mathfrak{m}_1$ iff θ_n is true on $\mathfrak{m}_2$.

Negation: ($\Rightarrow$) Assume that $\sim\phi$ is true on $\mathfrak{m}_1$. Then, by the definition of truth, $\sim$, ϕ is false on $\mathfrak{m}_1$. So, by RA, ϕ is false on $\mathfrak{m}_2$. It follows that $\sim\phi$ is true on $\mathfrak{m}_2$.

($\Leftarrow$) Assume that $\sim\phi$ is true on $\mathfrak{m}_2$. Then, by the definition of truth, $\sim$, ϕ is false on $\mathfrak{m}_2$. So, by RA, ϕ is false on $\mathfrak{m}_1$. It follows that $\sim\phi$ is true on $\mathfrak{m}_1$.

> Note that the ($\Leftarrow$) part is the same as the ($\Rightarrow$) part, but with $\mathfrak{m}_1$ and $\mathfrak{m}_2$ switched. For the rest of this proof we omit such redundant details.

Conditional: ($\Rightarrow$) Assume that $\theta_1 \rightarrow \theta_2$ is true on $\mathfrak{m}_1$. Then, by the definition of truth, $\rightarrow$, $\mathfrak{m}_1$ makes either θ_1 false or θ_2 true. (Case 1) $\mathfrak{m}_1$ makes θ_1 false. So, by RA, $\mathfrak{m}_2$ makes θ_1 false. (Case 2) $\mathfrak{m}_1$ makes θ_2 true. So, by RA, $\mathfrak{m}_2$ makes θ_2 true. In both cases $\mathfrak{m}_2$ makes $\theta_1 \rightarrow \theta_2$ true.

($\Leftarrow$) This is the same as ($\Rightarrow$), but with $\mathfrak{m}_1$ and $\mathfrak{m}_2$ switched.

Biconditional: ($\Rightarrow$) Assume that $\theta_1 \leftrightarrow \theta_2$ is true on $\mathfrak{m}_1$. Then, by the definition of truth, $\leftrightarrow$, $\mathfrak{m}_1$ either makes both θ_1 and θ_2 true or it makes both θ_1 and θ_2 false. (Case 1) $\mathfrak{m}_1$ makes θ_1 and θ_2 true. So, by RA, $\mathfrak{m}_2$ makes θ_1 and θ_2 true. (Case 2) $\mathfrak{m}_1$ makes θ_1 and θ_2 false. So, by RA, $\mathfrak{m}_2$ makes θ_1 and θ_2 false. In both cases $\mathfrak{m}_2$ makes $\theta_1 \leftrightarrow \theta_2$ true.

($\Leftarrow$) This is the same as ($\Rightarrow$), but with $\mathfrak{m}_1$ and $\mathfrak{m}_2$ switched.

Disjunction: ($\Rightarrow$) Assume that $(\theta_1 \vee \theta_2 \vee \ldots \vee \theta_n)$ is true on $\mathfrak{m}_1$. Then, by the definition of truth, $\vee$, there is at least one θ_i that is true on $\mathfrak{m}_1$, where

$1 \leq i \leq n$. So, by RA, θ_i is true on $\mathfrak{m}_2$. It follows that $(\theta_1 \vee \theta_2 \vee ... \vee \theta_n)$ is true on $\mathfrak{m}_2$.

($\Leftarrow$) This is the same as ($\Rightarrow$), but with $\mathfrak{m}_1$ and $\mathfrak{m}_2$ switched.

Conjunction: ($\Rightarrow$) Assume that $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$ is true on $\mathfrak{m}_1$. Then, by the definition of truth, $\wedge$, each θ_i is true on $\mathfrak{m}_1$, where $1 \leq i \leq n$. So, by RA, each θ_i is true on $\mathfrak{m}_2$. It follows that $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$ is true on $\mathfrak{m}_2$.

($\Leftarrow$) This is the same as ($\Rightarrow$), but with $\mathfrak{m}_1$ and $\mathfrak{m}_2$ switched.

Closure Step: These are all the ways of constructing SL sentences. Therefore, if $\mathfrak{m}_1$ and $\mathfrak{m}_2$ make the same assignments for all the sentence letters in some ϕ, then ϕ is true on $\mathfrak{m}_1$ iff ϕ is true on $\mathfrak{m}_2$.

■

2.5.3 Main Connective Theorem

We defined the main connective of a sentence ϕ as the connective that isn't in any proper subsentence of ϕ (2.11). Atomic sentences don't have any connectives, and so don't have main connectives. Most non-atomic sentences have just one main connective, but there is an exception: conjunctions (or disjunctions) with more than two conjuncts (disjuncts). For example, all three of the '$\wedge$' tokens in the following sentence are the main connectives: $(B \wedge A \wedge H \wedge G)$. Let's call sentences that have multiple tokens of '$\wedge$' as main connectives 'extended conjunctions', and those that have multiple tokens of '$\vee$' as main connectives 'extended disjunctions'. With this terminology established, we turn to another recursive proof.

Theorem 2.71. Main Connective Theorem: For every SL sentence ϕ, one of the following holds: (i) ϕ has no main connective; (ii) ϕ has exactly one main connective token; or (iii) ϕ is an extended conjunction (or disjunction) with $n-1$ main connective tokens, where n is the number of conjuncts (disjuncts).

Proof of Thm. 2.71, Main Connective Theorem:

Base Step: Every SL sentence of order 1 is just a sentence letter. Therefore it has no main connective.

Inheritance Step:

Recursive Assumption: Let ϕ and $\theta_1, \theta_2, ..., \theta_n$ be sentences of order k or less. Assume that one of (i)-(iii) holds of each of these sentences. Now consider the ways in which a sentence of order $k+1$ can be constructed from them:

Negation: Every connective in ϕ is in a proper subsentence of $\sim\phi$. The only connective not in a proper subsentence of $\sim\phi$ is '$\sim$'. So '$\sim$' is the main connective. Thus $\sim\phi$ has precisely one main connective, thereby meeting condition (ii).

Conditional/Biconditional: All the connectives in each of θ_1 and θ_2 is in a proper subsentence of $(\theta_1 \rightarrow \theta_2)$. The only connective not in a proper subsentence of $(\theta_1 \rightarrow \theta_2)$ is '$\rightarrow$'. So '$\rightarrow$' is the main connective. Thus $(\theta_1 \rightarrow \theta_2)$ has precisely one main connective, thereby meeting condition (ii). The same reasoning holds of $(\theta_1 \leftrightarrow \theta_2)$, except that it has '$\leftrightarrow$' as its main connective.

Conjunction/Disjunction: Consider a conjunction, $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$. If $n = 2$ then the reasoning of the 'Conditional/Biconditional' clause holds, and there is only 1 main connective. Otherwise $n > 2$, in which case the conjunction is extended. In that case, every connective in each of $\theta_1, \theta_2, ..., \theta_n$ is in a proper subsentence of $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$. The only connectives not in a proper subsentence are the '$\wedge$' tokens between each pair of θ_i and $\theta_i + 1$ subsentences, where $1 \leq i \leq n - 1$. So the main connectives are those $n - 1$ '$\wedge$' tokens. Thus $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$ has precisely $n - 1$ main connectives, thereby meeting condition (iii). The same reasoning holds of $(\theta_1 \vee \theta_2 \vee ... \vee \theta_n)$, except that it has $n - 1$ '$\vee$' tokens as its main connectives.

Closure Step: These are all the ways of constructing SL sentences. Therefore, every SL sentences has either no main connective, one main connective, or in the case of extended conjunctions or disjunctions with n conjuncts/disjuncts, $n - 1$ main connective tokens.

■

2.6 Disjunctive Normal Form

Consider the sentences below. Let's say we are given the truth values of their sentence letters. Which sentences are easier to evaluate? Which are more difficult?

9. $\sim(A \rightarrow \sim(C \wedge B)) \rightarrow (A \rightarrow C)$
10. $A \wedge R \wedge A \wedge \sim R$
11. $\sim(\sim(A \rightarrow \sim R) \rightarrow (A \rightarrow R))$
12. $(\sim A \vee \sim C \vee \sim B) \vee (\sim A \vee C)$

Clearly 10 and 12 are simpler to figure out than 9 and 11. In general, we find that the truth values of certain sentences are easier to calculate than others. Say that we want to figure out the truth value of a difficult sentence and we know that it's TFE to an easy sentence. Then if we compute the truth value of the easy one we know the value of the difficult one. This technique can be exploited to simplify the analysis of complicated SL sentences. In the above list of sentences, 9 is TFE to 12 and 10 is TFE to 11.

But how can we know whether some sentence is TFE to a simpler, more transparent sentence? It turns out that we can simplify a complicated sentence by a systematic series of steps, each of which replaces some subsentence with a simpler sentence that we know is TFE to the subsentence being replaced.

2.6.1 The TFE Replacement Theorem

We want a method of transforming a complicated sentence to an equivalent but simpler one. To that end we prove:

Theorem 2.72. Truth Functional Equivalence Replacement: Let ϕ be a subsentence of θ, ϕ and ϕ^* be TFE, and θ^* be the result of replacing one occurrence of ϕ by ϕ^* in θ. Then θ and θ^* are TFE.

Proof:

Base Step: Suppose that θ is an atomic sentence. Then $\theta = \phi$, since each atomic sentence has only itself as a subsentence. It also follows that $\theta^* = \phi^*$, since "substitution" is just replacement in this case. Since ϕ and ϕ^* are TFE, it follows that θ^* and θ are TFE.

Inheritance Step:

Recursive Assumption: Let ϕ and $\theta_1, \theta_2, \ldots, \theta_n$ be SL sentences of order k or less. And let ϕ^* and $\theta_1^*, \theta_2^*, \ldots, \theta_n^*$ be, respectively, SL sentences that are TFE to them. Now consider the ways in which a substitution can be made into a sentence of order $k+1$.

Negation: By RA, ϕ and ϕ^* are TFE. Then a model $\mathfrak{m}$ makes ϕ false iff it makes ϕ^* false. By the definition of truth, $\sim$, $\mathfrak{m}$ makes $\sim\phi$ true iff it makes ϕ false. Similarly, $\mathfrak{m}$ makes $\sim\phi^*$ true iff it makes ϕ^* false. Then $\mathfrak{m}$ makes $\sim\phi$ true iff it makes $\sim\phi^*$ true. Thus $\sim\phi$ and $\sim\phi^*$ are TFE.

Conditional, LHS Replacement: ($\Rightarrow$) Assume that model $\mathfrak{m}$ makes $(\theta_1 \to \theta_2)$ true. Then, by the definition of truth, $\to$, $\mathfrak{m}$ makes either θ_1 false or θ_2 true. θ_1 is to be replaced by θ_1^*. By RA, θ_1 is TFE to θ_1^*. So $\mathfrak{m}$ makes either θ_1^* false or θ_2 true. Then $\mathfrak{m}$ makes $(\theta_1^* \to \theta_2)$ true. By the definition of $\models$, $(\theta_1 \to \theta_2) \models (\theta_1^* \to \theta_2)$.

($\Leftarrow$) Assume that model $\mathfrak{m}$ makes $(\theta_1^* \to \theta_2)$ true. Then, by the definition of truth, $\to$, $\mathfrak{m}$ makes either θ_1^* false or θ_2 true. By RA, θ_1 is TFE to θ_1^*. So $\mathfrak{m}$ makes either θ_1 false or θ_2 true. Then $\mathfrak{m}$ makes $(\theta_1 \to \theta_2)$ true. By the definition of $\models$, $(\theta_1^* \to \theta_2) \models (\theta_1 \to \theta_2)$.

Thus, by theorem 2.50, $(\theta_1 \to \theta_2)$ is TFE to $(\theta_1^* \to \theta_2)$.

> Note that the ($\Leftarrow$) part is the same as the ($\Rightarrow$) part, but with θ_1 and θ_1^* switched. For the rest of this proof we omit such redundant details.

Conditional, RHS Replacement: ($\Rightarrow$) Assume that model $\mathfrak{m}$ makes $(\theta_1 \to \theta_2)$ true. Then, by the definition of truth, $\to$, $\mathfrak{m}$ makes either θ_1 false or θ_2 true. θ_2 is to be replaced by θ_2^*. By RA, θ_2 is TFE to θ_2^*. So $\mathfrak{m}$ makes either

θ_1 false or θ_2^* true. Then $\mathfrak{m}$ makes $(\theta_1 \to \theta_2^*)$ true. By the definition of $\models$, $(\theta_1 \to \theta_2) \models (\theta_1 \to \theta_2^*)$.

($\Leftarrow$) This is the same as ($\Rightarrow$), but with θ_2 and θ_2^* switched.

Thus, by theorem 2.50, $(\theta_1 \to \theta_2)$ is TFE to $(\theta_1 \to \theta_2^*)$.

Biconditional, LHS Replacement: ($\Rightarrow$) Assume that model $\mathfrak{m}$ makes $(\theta_1 \leftrightarrow \theta_2)$ true. Then, by the definition of truth, $\leftrightarrow$, $\mathfrak{m}$ either makes θ_1 and θ_2 both true or makes θ_1 and θ_2 both false. θ_1 is to be replaced by θ_1^*. By RA, θ_1 is TFE to θ_1^*. So $\mathfrak{m}$ either makes θ_1^* and θ_2 both true or makes θ_1^* and θ_2 both false. Then $\mathfrak{m}$ makes $(\theta_1^* \leftrightarrow \theta_2)$ true. By the definition of $\models$, $(\theta_1 \leftrightarrow \theta_2) \models (\theta_1^* \leftrightarrow \theta_2)$.

($\Leftarrow$) This is the same as ($\Rightarrow$), but with θ_1 and θ_1^* switched.

Thus, by theorem 2.50, $(\theta_1 \leftrightarrow \theta_2)$ is TFE to $(\theta_1^* \leftrightarrow \theta_2)$.

Biconditional, RHS Replacement: The proof for this case is left for the reader.

Conjunction: ($\Rightarrow$) Assume that model $\mathfrak{m}$ makes $(\theta_1 \wedge ... \wedge \theta_i \wedge ... \wedge \theta_n)$ true, where θ_i is the conjunct to be replaced. Keep in mind that i could also be 1 or n, so $1 \leq i \leq n$. Then, by the defition of truth, $\wedge$, each of $\theta_1, ..., \theta_i, ..., \theta_n$ is true on $\mathfrak{m}$. By RA, θ_i is TFE to θ_i^*. So, $\theta_1, ..., \theta_i^*, ..., \theta_n$ is true on $\mathfrak{m}$. Then $\mathfrak{m}$ makes $(\theta_1 \wedge ... \wedge \theta_i^* \wedge ... \wedge \theta_n)$ true. By the definition of $\models$, $(\theta_1 \wedge ... \wedge \theta_i \wedge ... \wedge \theta_n) \models (\theta_1 \wedge ... \wedge \theta_i^* \wedge ... \wedge \theta_n)$.

($\Leftarrow$) This is the same as ($\Rightarrow$), but with θ_i and θ_i^* switched.

Thus, by theorem 2.50, $(\theta_1 \wedge ... \wedge \theta_i \wedge ... \wedge \theta_n)$ and $(\theta_1 \wedge ... \wedge \theta_i^* \wedge ... \wedge \theta_n)$ are TFE .

Disjunction: The proof for this case is left for the reader.

Closure Step: Those are the only ways SL sentences can be formed; hence the theorem is proved.

■

Example 2.73. We can use this theorem to show that 9 and 12 above are equivalent, as are 10 and 11. Consider 9 and 12. (We leave 10 and 11 to the reader.) For any formulas ϕ and θ,

13. $(\phi \to \theta)$ and $(\sim\phi \vee \theta)$ are TFE (See ex. 2.62, on page 37)
14. $\sim\sim\phi$ and ϕ are TFE (See ex. 2.63, on page 37)
15. $\sim(\theta \wedge \phi)$ and $(\sim\theta \vee \sim\phi)$ are TFE (See 4 and 12, section 2.8.6)
16. $(\phi \vee \theta \vee \psi)$ and $(\phi \vee (\theta \vee \psi))$ are TFE (Obvious)

By making substitutions starting with 9 that the theorem says result in successive truth functionally equivalent sentences, we can get from 9 to 12 and thereby have shown that 12 is truth functionally equivalent to 9.

17. $\sim(A \to \sim(C \wedge B)) \to (A \to C)$ [9]

18. $\sim\sim(A\rightarrow\sim(C\wedge B))\vee(A\rightarrow C)$ [13]
19. $(A\rightarrow\sim(C\wedge B))\vee(A\rightarrow C)$ [14]
20. $(\sim A\vee\sim(C\wedge B))\vee(\sim A\vee C)$ [13]
21. $(\sim A\vee(\sim C\vee\sim B))\vee(\sim A\vee C)$ [15]
22. $(\sim A\vee\sim C\vee\sim B)\vee(\sim A\vee C)$ [16]

2.6.2 Disjunctive Normal Form

Sentences in disjunctive normal form are especially easy to evaluate.

Definition 2.74. A SL sentence is in *disjunctive normal form* (DNF) iff

(1) it contains no conditional ($\rightarrow$) or biconditional ($\leftrightarrow$),

(2) negations ($\sim$) only govern sentence letters, and

(3) no conjunction ($\wedge$) contains a disjunction ($\vee$) as a subsentence.

A typical example of a sentence in DNF is $(Q\wedge\sim R)\vee(\sim P\wedge R)$. The truth conditions for this sentence are transparent. The sentence $(Q\wedge\sim R)\vee(\sim P\wedge R)$ is true on a model $\mathfrak{m}$ when either $\mathfrak{m}(Q)=\mathtt{T}$ and $\mathfrak{m}(R)=\mathtt{F}$, or $\mathfrak{m}(P)=\mathtt{F}$ and $\mathfrak{m}(R)=\mathtt{T}$. Some less typical examples are:

23. Q
24. $\sim R$
25. $Q\wedge\sim R$
26. $\sim Q\vee R$

DNF is important because we can prove the following theorem.

Theorem 2.75. The Disjunctive Normal Form Theorem: Every sentence of SL is truth functionally equivalent to an SL sentence which is in DNF.

Proof: The proof relies on three lemmas, each of which can be established rigorously by recursive proof. We leave the details of these lemmas to the reader.

Here we provide a process showing how to turn any given sentence into one that's in DNF. We proceed in three stages, corresponding to the three lemmas that are necessary for a proof.

Step A: If a subsentence of ϕ has a conditional or biconditional as its main connective, i.e., is of the form $(\psi\rightarrow\theta)$ or $(\theta\leftrightarrow\psi)$, replace the subsentence by $(\sim\psi\vee\theta)$ or $(\psi\wedge\theta)\vee(\sim\psi\wedge\sim\theta)$ respectively. Repeat as necessary to obtain a sentence ϕ' without conditionals or biconditionals.

Step B:

(1) Replace any subsentence of the form $\sim\sim\psi$ in ϕ' with ψ.
(2) Replace any subsentence of the form $\sim(\psi \wedge \theta)$ in ϕ' with $(\sim\psi \vee \sim\theta)$.
(3) Replace $\sim(\psi \vee \theta)$ in ϕ' with $(\sim\psi \wedge \sim\theta)$.

Repeat as necessary to obtain ϕ'' in which negations govern nothing but sentence letters.

Step C: The only thing that could prevent ϕ'' from being in DNF is that some conjunctions govern some disjunctions, i.e., there is a subsentence $\theta \wedge (\psi_1 \vee \psi_2 \vee ... \vee \psi_n)$, or the reverse $(\psi_1 \vee \psi_2 \vee ... \vee \psi_n) \wedge \theta$. Those subsentences can be replaced by the equivalent $(\psi_1 \wedge \theta) \vee (\psi_2 \wedge \theta) \vee ... \vee (\psi_n \wedge \theta)$. Repeat as necessary.

■

A recursive proof would be more rigorous—it would have a clause for each SL connective, and would explain in each clause how to construct from each subsentence another TFE subsentence that is in DNF.

DNF sentences allow us see some of the advantages of formal languages. For instance, we can construct a simple, mechanical process that will tell us when a DNF sentence is TFF.

Let ϕ be some DNF sentence of the form $\theta_1 \vee \theta_2 \vee ... \vee \theta_n$. We can see that ϕ is TFF iff every disjunct θ_i is TFF. Because each θ_i is a conjunction with negated and unnegated sentence letters as the conjuncts, there is only one way that it can be TFF. A θ_i is TFF iff it has some sentence letter ψ as one conjunct and $\sim\psi$ as another.

It follows that we can use the following process to determine whether a DNF sentence ϕ is TFF. Check every disjunct of ϕ to see if it has some ψ and $\sim\psi$ as conjuncts. If so, then ϕ is TFF; otherwise it isn't. For example, consider the following DNF sentence:

$$(Q \wedge R \wedge \sim Q) \vee (\sim Q \wedge R \wedge R) \vee (O \wedge R \wedge \sim O)$$

The first disjunct, $(Q \wedge R \wedge \sim Q)$, is TFF because it has Q and $\sim Q$ as conjuncts; the third disjunct, $(O \wedge R \wedge \sim O)$, is also TFF. But the second disjunct, $(\sim Q \wedge R \wedge R)$, isn't TFF. So, the whole sentence isn't TFF.

If we were to replace the second disjunct with $(\sim Q \wedge R \wedge \sim R)$, so that the new whole sentence is:

$$(Q \wedge R \wedge \sim Q) \vee (\sim Q \wedge R \wedge \sim R) \vee (O \wedge R \wedge \sim O)$$

…then the result *is* TFF, because each disjunct has a sentence letter and its negation as conjuncts.

We now provide a mechanical method that determines whether *any* SL sentence ϕ is TFF. First, we use the process in 2.75 to construct an equivalent DNF sentence, ϕ^*. Then we use the method given above to determine whether ϕ^* is TFF. Because they

are equivalent, ϕ^* is TFF iff ϕ is TFF. No creativity is needed to apply this method—each individual step requires nothing more than following simple instructions.

We extend this method to determine whether any SL sentence is TFT. We take advantage of the fact that if you put a negation in front of a TFT sentence ϕ, the resulting sentence, $\sim\phi$, is TFF. So, all that is necessary to see whether ϕ is TFT is to negate ϕ, put $\sim\phi$ into DNF, and then to see whether the final result is TFF. If so, then ϕ is TFT. If not, then it isn't. As before, this process is entirely mechanical or *formal*. Note that, along with the truth table method in section 2.2.4, we now have two different formal methods for determining logical truth in SL. In later chapters our methods are closer to the DNF approach.

Finally, these two methods can be combined to make a mechanical test of whether some sentence ϕ is TFC. If ϕ is neither TFT nor TFF then it is TFC.

Any given sentence of SL is truth functionally equivalent to more than one DNF sentence. A sentence has DNFs that differ slightly for at least two reasons. First, for any sentence ϕ and sentence letter ψ, if ϕ is in DNF, then $\phi \vee (\psi \wedge \sim\psi)$ is TFE to ϕ and also in DNF. Second, sometimes a sentence in DNF can be simplified. Thus

27. $(Q \wedge R \wedge O) \vee (Q \wedge R \wedge N) \vee (Q \wedge R \wedge \sim O)$

can be simplified to the DNF

28. $(Q \wedge R) \vee (Q \wedge R \wedge N)$

and further to

29. $(Q \wedge R)$.

2.7 Truth Functional Expressiveness

The definition of truth in a model (def. 2.21, on page 23) associates each of the logical connectives of SL with a truth function.[20] We can think of the logical connectives of SL as truth functions—i.e., as having a meaning that's exhausted by the definition of truth. Do the five logical connectives of SL exhaust the range of possible logical connectives?

In one sense it's obvious they do not. For example, we could introduce a new connective, %, choose some number of places for it, and give some clause that describes how the truth value of a sentence with % as the main connective depends on the truth value of the component parts. As long as this clause differs from any of those in the definition of truth, % is distinct from the five in SL.

But though the five logical connectives of SL obviously do not exhaust all the possible connectives, there's still a sense in which they might indirectly cover them all. Even if % is distinct from all the connectives of SL, maybe there's still some sentence schema that is truth functionally equivalent to %, and that uses only (but

[20] See section 2.2.2 for more details.

not necessarily all of) the five connectives of SL. For example, say % is a 3-place connective, such that a sentence $\theta_1 \,\%\, \theta_2 \,\%\, \theta_3$ is true on a model $\mathfrak{m}$ iff at least two of the θ are true on $\mathfrak{m}$. So, the sentence is true if θ_1 and θ_2 are true, or if θ_1 and θ_3 are true, or if θ_2 and θ_3 are true. We can express this without '%', using $\wedge$ for 'and' and $\vee$ for 'or': $(\theta_1 \wedge \theta_2) \vee (\theta_1 \wedge \theta_3) \vee (\theta_2 \wedge \theta_3)$. Even though % is a connective that isn't in our language, we can use the connectives of SL to construct a truth functionally equivalent sentence.

A logical connective % is *definable* in terms of some set of other logical connectives Δ iff the connectives in Δ can be put together to define the same truth function as %. With this in mind, we can ask whether every logical connective is definable in terms of the five connectives of SL. We can also ask if any of the connectives of SL are definable in terms of the others. We answer the first of these questions with theorem 2.80, on this page. The second we consider now in the following examples.[21]

Example 2.76. Define conjunction using negation and disjunction.

Proof: Any sentence $\phi_1 \wedge ... \wedge \phi_n$ is TFE to the sentence $\sim(\sim\phi_1 \vee ... \vee \sim\phi_n)$. ■

Example 2.77. Define disjunction using negation and conjunction.

Proof: Any sentence $\phi_1 \vee ... \vee \phi_n$ is TFE to the sentence $\sim(\sim\phi_1 \wedge ... \wedge \sim\phi_n)$. ■

Example 2.78. Define conditional using negation and disjunction.

Proof: Any sentence $\phi_1 \rightarrow \phi_2$ is TFE to the sentence $\sim\phi_1 \vee \phi_2$. ■

Example 2.79. The pairs $\sim$ and $\wedge$, and $\sim$ and $\vee$ are each adequate to define the remaining connectives in SL.

Proof: By example 2.77, with $\sim$ and $\wedge$ we can define $\vee$. By example 2.62, $\phi \rightarrow \theta$ and $\sim\phi \vee \theta$ are TFE. So we can define $\rightarrow$ with $\sim$ and $\wedge$. As the reader can check, $\phi \leftrightarrow \psi$ is TFE to $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$. So we can define $\leftrightarrow$ with $\sim$ and $\wedge$.

We leave it to the reader to show that $\sim$ and $\vee$ are adequate to define the remaining connectives in SL. ■

We don't need all five connectives, but for the sake of convenience we keep them all. We also asked the following: Are the five operations we have enough? That is, are there other logical operations we can't express and should add notation for? It turns out that our connectives are adequate. There are no other logical operations we can't express. Although it does not strictly depend on the DNF theorem, the idea behind that theorem lets us prove this.

Theorem 2.80. The Truth-functional Expressive Completeness Theorem: Any truth-functional connective of any fixed number of arguments (ternary, quadernary, etc.) is already expressible in SL.

[21] See (Post 1921, Hodges 2001b: 17).

Proof: Any truth functional connective of a fixed number of arguments assigns T or F depending only on the values of the components, so it can be exactly described by a truth table. For example, consider the 4-place operation % given by truth table 2.2 below. We could attempt to find a complicated sentence in terms of various con-

ϕ_1	ϕ_2	ϕ_3	ϕ_4	$\%(\phi_1, \phi_2, \phi_3, \phi_4)$
T	T	T	T	T
T	T	T	F	F
T	T	F	T	T
T	T	F	F	F
T	F	T	T	F
T	F	T	F	T
T	F	F	T	F
T	F	F	F	F
F	T	T	T	T
F	T	T	F	F
F	T	F	T	F
F	T	F	F	F
F	F	T	T	F
F	F	T	F	F
F	F	F	T	T
F	F	F	F	F

Table 2.2: Truth Table for %

nectives that would express this, but it will be better for our purposes to construct a DNF equivalent systematically. We know from the first line that the expression $\%(\phi_1, \phi_2, \phi_3, \phi_4)$ is true when all components are, that is, if $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4)$ is true; we know from the third line it is true when the first two, ϕ_1 and ϕ_2, and fourth, ϕ_4, are true and the third, ϕ_3, false, i.e., $(\phi_1 \wedge \phi_2 \wedge \sim\phi_3 \wedge \phi_4)$. We also know it is true when only the first, ϕ_1, and third, ϕ_3, are true, i.e., $(\phi_1 \wedge \sim\phi_2 \wedge \phi_3 \wedge \sim\phi_4)$, when the first, ϕ_1, is false and the other three, ϕ_2, ϕ_3, and ϕ_4, are true i.e., $(\sim\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4)$ and when all but the fourth, ϕ_4, are false, i.e., $(\sim\phi_1 \wedge \sim\phi_2 \wedge \sim\phi_3 \wedge \phi_4)$. Because each of these conjunctions is true exactly when the corresponding line is true the whole sentence will be true when any one of them is true, i.e., it is equivalent to the formula:

30. $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4) \vee (\phi_1 \wedge \phi_2 \wedge \sim\phi_3 \wedge \phi_4) \vee (\phi_1 \wedge \sim\phi_2 \wedge \phi_3 \wedge \sim\phi_4) \vee$
$(\sim\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4) \vee (\sim\phi_1 \wedge \sim\phi_2 \wedge \sim\phi_3 \wedge \phi_4)$

We could simplify this sentence further, but that is not important for our purposes. We now can see how to read off from any truth table for any operation on any fixed number of sentences a DNF representation that is equivalent. Our five connectives are enough. ■

Since either of the pairs '$\sim$' and '$\wedge$', or '$\sim$' and '$\vee$' are adequate to define the remaining connectives in SL (see ex. 2.76, on page 50) we can see that either of those pairs is adequate to define all truth-functional connectives. We can even improve on that though, because there are two connectives either of which would be adequate all by itself. One is the Sheffer stroke, named after the logician who first demonstrated its properties and the symbol he used, '|', but it is sometimes called NAND. Its definition is that $(\phi_1|\phi_2|\ldots|\phi_n)$ is true iff at least one component is false. So, $(\phi_1|\phi_1)$ is equivalent to $\sim\phi$. And $(\sim\phi_1|\sim\phi_2|\ldots|\sim\phi_n)$ is true just in case at least one component is false. That means at least one ϕ_i is true, which is to say that the sentence is equivalent to a disjunction. The other connective that is adequate by itself is NOR, which is defined to be true iff all the components are false. It is left to the reader as an optional exercise to show how to define the other connectives using NOR.

Thus, to get a language just as expressive as SL, we only need one logical connective (either NAND or NOR), not five. If we had a taste for cutting down basic symbols, we could go further and generate an infinite set of sentence letters by using just one symbol, 'A', and generating new sentence letters by concatenating prime marks '$'$' to it. Then all we need are parentheses (though there are ways to do without these too). Such a language is sparse, but it is just as expressive as SL. Most people would find such a language difficult to work with, but computers love them.

2.8 Exercises

2.8.1 Recursive Definition Problems

1. Although it's not framed as one, definition 2.9 (on page 19) of order is a recursive definition. Rewrite it so that the base, generating, and closure clauses are explicit.

2. Although we don't give a recursive definition, a recursive definition can be given for the unofficial SL sentences. (Definition 2.4, on page 17 is the definition we give.) Write down a recursive definition for unofficial SL sentences.

2.8.2 Construction Trees

Write the construction tree for each of the following SL sentences.

1. $\sim\sim\sim B$
2. $\sim(B \vee (A \rightarrow A))$
3. $(\sim B \vee (A \rightarrow A))$
4. $(((A \wedge B) \leftrightarrow A) \rightarrow \sim(B \rightarrow C))$
5. $(((A \leftrightarrow B) \wedge A) \rightarrow \sim(B \rightarrow C))$
6. $(P \wedge (Q \wedge R))$
7. $((P \wedge Q) \wedge R)$
8. $((P \wedge \sim R) \rightarrow \sim Q)$
9. $(P \wedge (\sim R \rightarrow \sim Q))$
10. $\sim((P \rightarrow Q) \vee (P \rightarrow Q))$

2.8.3 Official and Unofficial Sentences

Which of these are official sentences? Which are unofficial? Which are neither official nor unofficial sentences? If neither, how could you make it either an official or unofficial sentence? Note: there might be multiple different ways to make it an official or unofficial sentence. Finally, if it's a sentence (official or unofficial), then give its order and the number of subsentences in it.

1. $(A \to B \wedge C)$
2. $(A \to (B \to C))$
3. $A \to (B \wedge C \wedge B)$
4. $(A \to (\theta \wedge C))$
5. $(A \to (B \wedge C \vee D))$
6. $(A \to (Z \wedge C))$
7. $(\sim A \to (B_{374} \wedge C))$
8. $(A \to (B \wedge C))$
9. $(\sim(B \wedge C))$
10. $(B \wedge \sim\sim M \wedge D)$

2.8.4 Truth in a Model

Consider the model $\mathfrak{m}_1$ such that $\mathfrak{m}_1(A) = \mathsf{T}$, $\mathfrak{m}_1(B) = \mathsf{F}$, $\mathfrak{m}_1(C) = \mathsf{T}$, $\mathfrak{m}_1(D) = \mathsf{F}$, and $\mathfrak{m}_1(E) = \mathsf{T}$; and the model $\mathfrak{m}_2$ such that $\mathfrak{m}_2(A) = \mathsf{T}$, $\mathfrak{m}_2(B) = \mathsf{T}$, $\mathfrak{m}_2(C) = \mathsf{F}$, $\mathfrak{m}_2(D) = \mathsf{F}$, and $\mathfrak{m}_2(E) = \mathsf{F}$. Give the truth values of each of the following SL sentences on each of these two models.

1. $(A \vee B) \to (C \wedge D)$
2. $(A \vee B) \to (C \wedge \sim D)$
3. $(C \wedge D) \to (A \vee B)$
4. $(A \to C) \wedge E$
5. $\sim(B \to E) \wedge D$
6. $\sim(A \vee B) \vee (A \wedge B \wedge (E \to E))$
7. $A \vee (B \to (E \to E))$
8. $(A \wedge E) \vee (\sim D \wedge C) \vee (B \wedge \sim A)$

2.8.5 TFT, TFF, and TFC

For each of the following say whether the sentence is TFC, TFF or TFT. If it is TFC, give a model which makes the sentence true and another model which makes it false. If it is TFF, justify your answer without truth tables (i.e., explain why there is no model which makes the sentence true). If it is TFT, again justify your answer without truth tables (i.e., explain why every model makes the sentence true).

1. $(A \to B) \to (\sim B \vee \sim A)$
2. $(A \wedge B) \to (A \leftrightarrow B)$
3. $(\sim A \vee \sim B) \to \sim(A \wedge B)$
4. $A \vee (A \to B)$
5. $\sim(A \vee B) \to (\sim A \wedge \sim B)$
6. $\sim(A \leftrightarrow B) \to (\sim A \leftrightarrow B)$
7. $(A \wedge (B \vee C)) \to ((A \wedge B) \vee C)$
8. $\sim A \to (A \to B)$

9. $(\sim A \wedge \sim B) \rightarrow \sim(A \vee B)$
10. $A \rightarrow (A \rightarrow B)$
11. $(A \leftrightarrow B) \rightarrow (A \wedge B)$
12. $\sim(A \rightarrow B) \rightarrow A$
13. $\sim(A \wedge B) \rightarrow (\sim A \vee \sim B)$
14. $A \rightarrow (B \rightarrow A)$
15. $(A \rightarrow B) \rightarrow (A \wedge \sim B)$
16. $\sim(A \vee (A \rightarrow B))$

2.8.6 Entailment Problems for SL

For each of the following, without using truth tables show whether the entailment holds.

> There are a number of different methods for thinking through these problems. Remember that an entailment means that on all $\mathfrak{m}$ if the LHS is true then the RHS is also true. One approach is to show that making the LHS true forces the RHS to be true. Another is to show that making the RHS false forces the LHS to be false. Both are examples of arguing that it is not possible for the LHS to be true and the RHS false. Another method is showing that all $\mathfrak{m}$ either make the LHS false or the RHS true. If the entailment does not hold, you can show this by providing a counterexample.

1. $\sim A \models (A \rightarrow B)$
2. $(A \rightarrow B) \models (\sim B \vee \sim A)$
3. $(\sim A \wedge \sim B) \models \sim(A \vee B)$
4. $(\sim A \vee \sim B) \models \sim(A \wedge B)$
5. $A \models (A \rightarrow B)$
6. $\sim(A \leftrightarrow B) \models (\sim A \leftrightarrow B)$
7. $(A \wedge (B \vee C)) \models ((A \wedge B) \vee C)$
8. $\sim(A \vee B) \models (\sim A \wedge \sim B)$
9. $A \models (B \rightarrow A)$
10. $(A \wedge B) \models (A \leftrightarrow B)$
11. $\sim(A \rightarrow B) \models (B \rightarrow A)$
12. $\sim(A \wedge B) \models (\sim A \vee \sim B)$
13. $\sim(A \rightarrow B) \models A$
14. $(A \leftrightarrow B) \models (A \wedge B)$

2.8.7 More Entailment Problems for SL

Show whether, for any SL sentence ϕ, θ, and ψ, each of the following statements is true.

1. $\models (\phi \rightarrow \theta) \vee (\theta \rightarrow \phi)$
2. Either $\models (\phi \rightarrow \theta)$, or $\models (\theta \rightarrow \phi)$
3. If $\models (\phi \rightarrow \theta)$ and $\models \phi$, then $\models \theta$
4. If $\models (\phi \rightarrow \theta)$ and $\models \sim\phi$, then $\models \sim\theta$

5. $\models (\phi \rightarrow \theta) \vee (\theta \rightarrow \psi)$
6. If $(\phi \wedge \theta) \models \psi$, then both $\phi \models \psi$ and $\theta \models \psi$
7. If $\psi \models (\phi \wedge \theta)$, then both $\psi \models \theta$ and $\psi \models \phi$

2.8.8 Even More Entailment Problems: Truth-preservation Lemma

Show that, for any SL sentence ϕ, θ, and ψ, each of the following entailments holds. Showing that these entailments hold will be helpful later, since they are needed in the proof of theorems 8.2 (on page 179) and 6.8 (on page 152).

1. $\phi \models \phi$.
2. $\theta \rightarrow \psi, \theta \models \psi$
3. $\theta \wedge \psi \models \psi$
4. $\theta \wedge \psi \models \theta$
5. $\theta, \psi \models \theta \wedge \psi$
6. $\theta \vee \psi, \theta \rightarrow \phi, \psi \rightarrow \phi \models \phi$
7. $\theta \models \theta \vee \psi$
8. $\theta \rightarrow (\psi \wedge \sim\psi) \models \sim\theta$
9. $\sim\theta \rightarrow (\psi \wedge \sim\psi) \models \theta$
10. $\theta \rightarrow \psi, \psi \rightarrow \theta \models \theta \leftrightarrow \psi$
11. $\theta \leftrightarrow \psi, \psi \models \theta$
12. $\theta \leftrightarrow \psi, \theta \models \psi$
13. $\psi \rightarrow \theta, \sim\theta \models \sim\psi$
14. $\psi \vee \theta, \sim\theta \models \psi$
15. $\theta \vee \psi, \sim\psi \models \theta$
16. $\theta, \sim\theta \models \psi$
17. $\psi \leftrightarrow \theta \models \sim\psi \leftrightarrow \sim\theta$

2.8.9 Truth Functional Equivalence

Without using truth tables, show that each of the following pairs of sentences is TFE, for any sentences got by substituting into the schemas. Showing that these pairs are TFE will be helpful later, since it's both needed for the proof of theorem 6.8 (on page 152) and it amounts to proving theorem 6.11 (on page 153) (including for $\leftrightarrow$-*Exchange*), which is needed to prove theorem 6.10 (on page 153).

1. $\sim(\phi_1 \wedge ... \wedge \phi_n)$, $\sim\phi_1 \vee ... \vee \sim\phi_n$
2. $\sim(\phi_1 \vee ... \vee \phi_n)$, $\sim\phi_1 \wedge ... \wedge \sim\phi_n$
3. $\sim\sim\phi$, ϕ
4. $\phi \rightarrow \theta$, $\sim\phi \vee \theta$
5. $\phi \rightarrow \theta$, $\sim\theta \rightarrow \sim\phi$
6. $\sim(\phi \rightarrow \theta)$, $\phi \wedge \sim\theta$
7. $\theta \wedge (\phi_1 \vee ... \vee \phi_n)$, $(\theta \wedge \phi_1) \vee ... \vee (\theta \wedge \phi_n)$

8. $(\phi_1 \vee ... \vee \phi_n) \wedge \theta$, $(\phi_1 \wedge \theta) \vee ... \vee (\phi_n \wedge \theta)$
9. $\theta \vee (\phi_1 \wedge ... \wedge \phi_n)$, $(\theta \vee \phi_1) \wedge ... \wedge (\theta \vee \phi_n)$
10. $(\phi_1 \wedge ... \wedge \phi_n) \vee \theta$, $(\phi_1 \vee \theta) \wedge ... \wedge (\phi_n \vee \theta)$
11. $\theta \leftrightarrow \psi$, $(\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)$

2.8.10 Relations Between SL Sentences

What relations hold among these sentences? Specifically, say whether they are contradictory, contrary, subcontrary, independent, or truth functionally equivalent. You need to supply 30 answers for each problem: for each sentence ϕ, you must determine for each of the 5 relations whether it holds between ϕ and each of the 6 other sentences.

1. A
2. $A \wedge B$
3. $\sim A \wedge B$
4. $A \rightarrow C$
5. $A \rightarrow \sim C$
6. $(A \wedge B) \vee C$
7. $D \wedge \sim D$

2.8.11 Recursive Proofs

1. Prove that all positive multiples of 10 are also multiples of 5.
2. For each even positive integer n, prove that dividing n by 2 results in another positive integer.

2.8.12 SL Recursive Proofs

Prove each of the following claims using a recursive proof.

1. In every SL sentence which is TFF, there is a subsentence which is TFC.

   > *Hint:* The idea behind this proof is easy. Don't over-think it.

2. Show that every TFF sentence of SL contains at least one '$\sim$'.

   > *Hint:* To prove this, it is useful to prove a more specific statement, namely that if ϕ is an SL sentence that does not contain any negations then ϕ is true on the model that assigns true to all sentence letters.

3. For every sentence ϕ of SL, the number of left parentheses occurring in ϕ is less than the number of subsentences. In other words, if LPϕ is the number of left parentheses in ϕ and SSϕ is the number of subsentences, then LPϕ $<$ SSϕ.

4. The number of subsentences in any official SL sentence ϕ is equal to: the number of tokens of sentence letters in ϕ plus the number of tokens of negation in ϕ plus the number of tokens of left parentheses in ϕ.

5. For every sentence ϕ of SL, there is a TFE sentence ϕ' without conditionals or biconditionals.

2.8.13 DNF

Put the following into disjunctive normal form.

1. $\sim(Q \to R) \vee (Q \to \sim R)$
2. $O \wedge (O \to Q)$
3. $((Q \to R) \to Q) \to Q$
4. $\sim(Q \to R) \wedge (O \vee P)$

Chapter 3

Quantifier Language I

3.1 The Language QL1

3.1.1 Sentences of QL1

SL allows us to investigate certain aspects of logical consequence, but leaves out a great deal of interest. The 'atoms' of SL are sentence letters, which are each assigned either `T` or `F` by a model. Sentence letters can only represent declarative sentences of the English language, which means that SL is too coarse-grained to capture the logical meaning of certain *parts* of a sentence. For example:

1. All women are mortal.
2. Ophelia is a woman.

   Therefore,
3. Ophelia is mortal.

The third sentence is a logical consequence of the first two, but we cannot use SL to represent this as an entailment. Let sentence letter A stand for 'All women are mortal,' let B stand for 'Ophelia is a woman,' and let C stand for 'Ophelia is mortal'. The entailment claim $A, B \models C$ does not hold, because there is a model $\mathfrak{m}$ that assigns `T` to A and B, but assigns `F` to C. We need a formal language more expressive than SL.

This new language needs symbols that represent named objects, including people like Ophelia. It also needs symbols that stand for the predicates of English. Predicates correspond roughly to what you get if you take an English sentence and remove the subject, leaving a blank, e.g.:

4. 'John is tall' ⇒ '_____ is tall'
5. 'Ophelia is a woman' ⇒ '_____ is a woman'
6. 'Ophelia is mortal' ⇒ '_____ is mortal'

To apply a predicate without invoking a name we use a variable, which functions somewhat like a pronoun in English. And to account for the word 'all' in our new language, we use a *quantifier*. (We discuss quantifiers a bit later.) In this chapter we

outline a formal language with these features, and thus which can capture the kind of logical consequence exhibited above.

Many logic texts call the following language PL, for *predicate language*. We use 'QL' for *quantifier language*, because the quantifiers are more important than the predicates. However, before turning to the full language of QL (in the next chapter), we first consider a simpler sublanguage which we call 'QL1'. We call it 'QL1' because the predicates will all be 1-place.[1]

QL1 has all the basic symbols of SL, plus a few more.

Definition 3.1. The *basic symbols* of QL1 are:

(1) Logical Connectives: those of SL, plus $\forall$ and $\exists$

(2) Punctuation Symbols: those of SL

(3) Sentence Letters: those of SL

(4) Individual Constants: a, b, c, d, ..., p, a_1, b_1, c_1, ..., p_1, a_2, ...

(5) Individual Variables: u, v, w, x, y, z, u_1, v_1, ..., z_1, u_2, ...

(6) 1-Place Predicates: A', B', ..., T', A'_1, B'_1, ..., T'_1, A'_2, B'_2, ...

The most prominent additions are the new logical connectives, the two quantifiers. The first, '$\forall$', is called the *universal quantifier*. It corresponds to the words 'all' or 'every' in English. The second, '$\exists$', is called the *existential quantifier*. It corresponds to 'there exists', 'there is', or 'some', as in 'Some elephants live a long time.'[2]

The individual constants of QL correspond roughly to names in English. They are lowercase Roman letters that start at 'a' and stop at 'p', and then they start at 'a' again with subscripted integers. So we have, for example, a_1, a_2, a_3, and so on.

Next, the individual variables correspond roughly to pronouns in English. They are Roman lowercase letters that go from 'u' to 'z' and then start at 'u' again with subscripted positive integers, e.g. 'u_1'.

We also have 1-place predicates in QL1. The one-place predicates are capital Roman letters going from 'A'' to 'T'' and then starting from 'A'' again with subscripted integers, e.g. 'A'_1'. There are an infinite number of each of the individual constants, variables, and 1-place predicates. This is so we never run out of QL1 symbols when analyzing some sentence or argument, no matter how complex or long it is.

3.1.2 Formulas of QL1

Before we can define *sentences* of QL1 we must first define a larger set of strings called 'formulas.' But before that we must introduce another kind of metavariable to

[1] In QL there will be many-place predicates. The basics of a less formal version of QL1 were developed by Aristotle over 2,000 years before Gottlob Frege and others developed the full language we're calling QL. QL was a big step for mankind.

[2] Although Frege, Peirce, and Mitchell first introduced quantifiers, the notation used here comes from Russell, who Church (1956: 288) says modified Peano's notation.

MathEnglish, one that ranges over for QL1 individual variables.[3] We use lowercase Greek letters to stand for variables of QL1, usually but not necessarily always from the beginning of the Greek alphabet (e.g., 'α' and 'β'). Although we also used lowercase Greek letters as variables for SL sentences and will continue to do so for QL1 sentences and formulas, confusion shouldn't arise as we typically use 'ϕ', 'ψ', and 'θ' for SL and QL1 sentences and use 'α' and 'β' for QL1 variables.

Definition 3.2. The *formulas of QL1* are given by the following recursive definition:

Base Clauses:

(1) A sentence letter is a formula.
(2) A 1-place predicate followed by one individual constant or variable is a formula.

Generating Clauses:

(1) If ϕ is a formula then so is $\sim\phi$.
(2) If ϕ and θ are formulas then so are $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.
(3) If all of $\phi_1, \phi_2, \phi_3, \phi_4, \ldots, \phi_n$ are formulas (where n is an integer ≥ 2) then so are $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \ldots \vee \phi_n)$.
(4) If α is a QL1 variable and ϕ is a formula that does not contain an expression of the form $\forall\alpha$ or $\exists\alpha$, then $\forall\alpha\phi$ and $\exists\alpha\phi$ are formulas.

Closure Clause: A string of symbols is a formula only if it can be generated by the clauses above.

For example, $A'\mathrm{b}$ is a formula, and so is $D'x_4$. Each of these formulas is atomic. Formulas of the form $\forall\alpha\phi$ are called *universal* formulas and formulas of the form $\exists\alpha\phi$ are called *existential*. Here are some additional examples of QL1 formulas.

1. $\forall x J'x$
2. $\sim\exists y K'x$
3. $\exists z L'_{12}\mathrm{b}$
4. $\exists y\sim\forall x G'y$
5. $\forall x\exists y H'z$
6. $(\forall x\forall z P'z_{176} \rightarrow \forall y G'y)$

Contrarily, $\forall x\forall x G'x$ is *not* a formula. That's because it's of the form $\forall x\phi$ where ϕ is a formula that contains the expression $\forall x$.[4] Neither is $\forall\mathrm{a}G'\mathrm{a}$, because $\forall$ *must* be

[3] MathEnglish variables are symbols of the metalanguage while individual variables of QL1 are symbols of the object language.

[4] Continuing the practice started in section 2.1.3, we do not always put symbols, expressions, and sentences of QL1 that are mentioned (instead of used) in single quotes. For example, the tokens of the universal and existential quantifiers in definition 3.12 (on page 65) should, strictly speaking, be in quotes because they *mention* the symbols. Being stringent in the use of single quotes—and overly sensitive to the use/mention distinction in general—can cloud what are relatively clear and straightforward concepts. Wherever it may be helpful, we provide footnotes with more rigorous and detailed explanation.

paired with a variable, but 'a' is a constant.

Finally, we have unofficial formulas, just as we had unofficial sentences in SL (compare with def. 2.4, on page 17).

Definition 3.3. A string of symbols is an *unofficial* formula iff we can obtain it from an official formula by

(1) deleting outer parentheses,

(2) replacing official parentheses ( ) with square brackets [ ] or curly brackets { }, or

(3) omitting primes ′ on a predicate letter.

From an unofficial formula we can unambiguously reconstruct the corresponding official formula.

3.1.3 Other Properties of Formulas

As in section 2.1.4 for sentences of SL, we define the concepts of subformula, order, main connective, and construction tree for formulas of QL1.

Definition 3.4. The following clauses define when one formula is a *subformula* of another:

(1) Every formula is a subformula of itself.

(2) ϕ is a subformula of $\sim\phi$.

(3) ϕ and θ are subformulas of $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.

(4) Each of $\phi_1, \phi_2, \ldots, \phi_n$ is a subformula of $(\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \ldots \vee \phi_n)$.

(5) ϕ is a subformula of $\forall\alpha\phi$ and $\exists\alpha\phi$.

(6) (Transitivity) If ϕ is a subformula of θ and θ is a subformula of ψ, then ϕ is a subformula of ψ.

(7) That's all.

Importantly, the quantifier phrase is *not* a subformula. Thus, $\forall x \forall z G'x$ has three subformulas: $\forall x \forall z G'x$, $\forall z G'x$, and $G'x$. Neither $\forall x$ nor $\forall x \forall z$ is a subformula. Middle parts of the formula cannot be removed to form a subformula. So $\forall x G'x$ isn't one either.

Definition 3.5. The *order* of a formula is defined parallel to that of an SL sentence (see def. 2.9, on page 19) with extra clauses specifying that adding a quantifier increases the order by one. The following clauses define the *order* of a formula. Let $\mathrm{ORD}\phi$ be the order of ϕ. Then:

(1) If ϕ is a sentence letter then $\mathrm{ORD}\phi = 1$.

(2) If P' is a predicate letter and α is a constant or variable then ORD$P'\alpha = 1$.

(3) For any formula ϕ, ORD$\sim\phi$ = ORD$\phi + 1$.

(4) For any formulas ϕ and θ, ORD$(\phi \rightarrow \theta)$ is one greater than the max of ORDϕ and ORDθ. Likewise, ORD$(\phi \leftrightarrow \theta)$ is one greater than the max of ORDϕ and ORDθ.

(5) For any formulas $\phi_1, \ldots, \phi_n$, ORD$(\phi_1 \wedge \ldots \wedge \phi_n)$ is one greater than the max of ORDϕ_1, ..., ORDϕ_n.

(6) For any formulas $\phi_1, \ldots, \phi_n$, ORD$(\phi_1 \vee \ldots \vee \phi_n)$ is one greater than the max of ORDϕ_1, ..., ORDϕ_n.

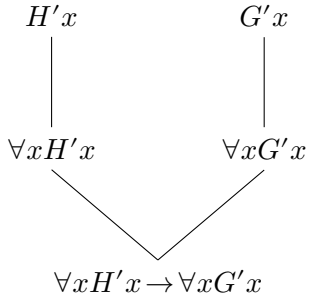
(7) For formulas $\forall\alpha\phi$ and $\exists\alpha\phi$, ORD$\forall\alpha\phi$ = ORD$\exists\alpha\phi$ = ORD$\phi + 1$.

(8) That's all.

Definition 3.6. The *main connective* of a formula is the connective token (or tokens) that occur(s) in the formula but in no proper subformula.

Definition 3.7. The *construction tree* of a formula is defined as with SL (def. 2.13, on page 20) but with the obvious extension for quantifiers. The order of a formula is the height of the construction tree's longest branch, measured by counting nodes. Each node in the tree (including the bottom node) is a subformula, and the main connective is the connective added at the very bottom of the tree.

Example 3.8. Consider the (unofficial) formula $\forall xH'x \rightarrow \forall xG'x$. Its construction tree is:



The height of the construction tree is 3, so that's the order of the formula. It has five subformulas:

(1) $\forall xH'x \rightarrow \forall xG'x$

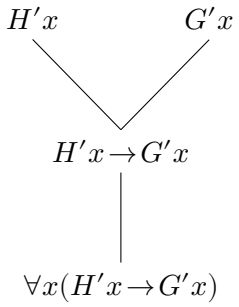
(2) $\forall xH'x$

(3) $\forall xG'x$

(4) $H'x$

(5) $G'x$

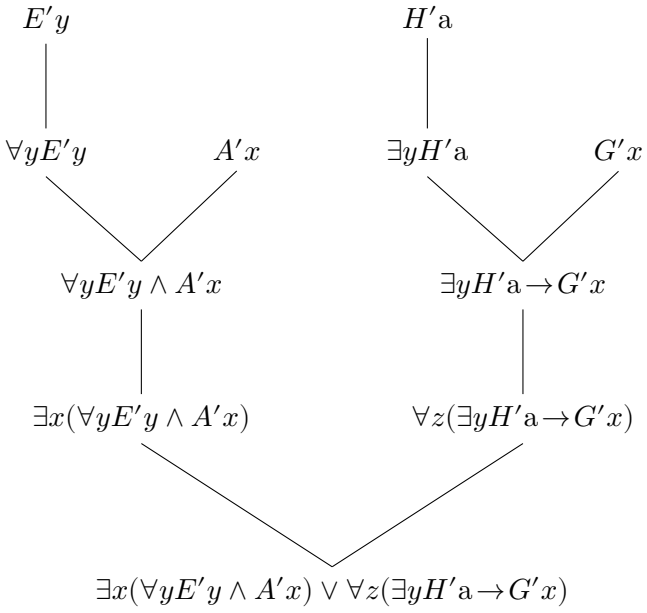
Example 3.9. Next consider the formula $\forall x(H'x \to G'x)$. It is not to be confused with the formula from example 3.8. Consider carefully the differences between the two. The formula in 3.8 has '$\to$' as its main connective, but the main connective of this formula is '$\forall$'. Its construction tree is:



The height of the tree is 3 so the order of the formula is 3. It has four subformulas:

(1) $\forall x(H'x \to G'x)$
(2) $(H'x \to G'x)$
(3) $H'x$
(4) $G'x$

Example 3.10. Consider the formula $\exists x(\forall y E'y \wedge A'x) \vee \forall z(\exists y H'\mathrm{a} \to G'x)$. Its construction tree is:



The height of the construction tree is 5, so that's the order of the formula. It has eleven subformulas:

(1) $\exists x(\forall y E'y \wedge A'x) \vee \forall z(\exists y H'\mathrm{a} \to G'x)$

(2) $\exists x(\forall y E'y \wedge A'x)$
(3) $\forall y E'y \wedge A'x$
(4) $\forall y E'y$
(5) $A'x$
(6) $E'y$
(7) $\forall z(\exists y H'\mathrm{a} \rightarrow G'x)$
(8) $\exists y H'\mathrm{a} \rightarrow G'x$
(9) $\exists y H'\mathrm{a}$
(10) $G'x$
(11) $H'\mathrm{a}$

3.1.4 Sentences of QL1

QL1 sentences are defined in terms of QL1 formulas, but we need a few other definitions before we can proceed: *scope*, and *free* vs. *bound* variables.

Definition 3.11. In a formula $\exists\alpha\phi$ or $\forall\alpha\phi$, we say that ϕ is the *scope* of the quantifier $\exists\alpha$ or $\forall\alpha$.

For example, in the formula $\forall y E'y$, $E'y$ is the scope of $\forall y$. Note that the scope of a quantifier may be only a small subformula of a much larger whole. Consider $\exists x(\forall y E'y \wedge A'x)$. The scope of $\forall y$ is just $E'y$, but the scope of $\exists x$ is $(\forall y E'y \wedge A'x)$.

Definition 3.12. A variable token α in a formula ϕ is *bound* iff either (i) it's part of a quantifier expression, $\exists\alpha$ or $\forall\alpha$; or (ii) it occurs within the scope of a quantifier expression with variable α.

Definition 3.13. A variable token in a formula ϕ is *free* iff it is not bound.

In the formula $\exists x(G'x \wedge H'x)$, the first token of x is bound because it's part of the quantifier expression '$\exists x$'. The second and third tokens of x are bound because they are within the scope of '$\exists x$'.[5]

Now we are ready for the definition of a QL1 sentence:

Definition 3.14. A string of QL1 symbols is a *sentence* iff it is a formula that contains no free variables.

Definition 3.15. An *atomic sentence* of QL1 is an atomic formula of QL1 that has no free variable.

We have unofficial sentences too, just as we have unofficial formulas.

Definition 3.16. A string of symbols is an *unofficial* sentence iff it's an unofficial formula that contains no free variables. In other words, we can get an unofficial sentence from an official one by

(1) deleting outer parentheses,

[5] We can also determine whether a variable is bound by thinking of a tree of the formula. A variable token is bound iff a quantifier with the same variable appears below or at the same level as (but still on the same branch as) that token. The quantifier that binds a variable is the *first* quantifier that appears below or at the same level as (but still on the same branch as) the variable.

(2) replacing official parentheses ( ) with square brackets [ ] or curly brackets { }, or

(3) omitting primes ′ on the predicate letters.

Example 3.17. Both formulas $\forall x H'x \to \forall x G'x$ and $\forall x(H'x \to G'x)$ from examples 3.8 and 3.9 are sentences, because in each all variables are bound. Below we have arrows pointing from each variable token to the quantifier that binds it. (There are no arrows for the variables in quantifier expressions since it's obvious which quantifiers they belong to.)

(1) $\forall x H'x \to \forall x G'x$

(2) $\forall x(H'x \to G'x)$[6]

Example 3.18. The formula $\exists x(\forall y E'x \wedge A'x) \vee \forall z(\exists y H'\mathrm{a} \to G'x)$ is not a sentence. There is a free variable in it. The free variable is underlined, while arrows point from each bound variable to the quantifier that binds it (again, ignoring variables in quantifier expressions).

(1) $\exists x(\forall y E'x \wedge A'x) \vee \forall z(\exists y H'\mathrm{a} \to G'\underline{x})$

The two quantifiers on the RHS of the disjunction are not binding any token variables, besides the ones that appear in the quantifier expressions themselves.

Example 3.19. In each of the following three formulas the free variable tokens are underlined, and arrows go from variable tokens to the quantifiers that bind them.

(1) $\forall x(B \wedge \exists z K'x) \to \exists x N x$

(2) $D \wedge (\exists u \forall w P'u \vee C'\underline{y})$

(3) $\exists z((H'\underline{x} \wedge G) \leftrightarrow Dz)$

Formula (1) has no free variables but (2) and (3) do, so (1) is a sentence but (2) and (3) aren't.

[6] You can also use trees to show that these variables are all bound. Look at the construction trees of the formulas from examples 3.8 and 3.9. You will see that in each case the indicated quantifier is the first that appears below, but still on the same branch as, the token variable.

3.2 Models

As with SL, sentences of QL1 have no inherent meaning. But, also as with SL, we give 'models' for sentences of QL1. These models allow us to define and investigate entailment for QL1.

Our goal is to define 'model' such that each QL1 sentence has a determinate truth value on each model. Note that only sentences of QL1 have true values, and not formulas. A formula that isn't a sentence has some free variable. Accordingly, it corresponds to a grammatical sentence of English that doesn't have a determinate truth value, because it contains one or more pronouns. For example, the sentence 'He is the author of Waverley,' may be either true or false, depending on who 'he' is. In ordinary conversation we use context to determine the referent of a pronoun. Someone discussing the English author Sir Walter Scott may read the above sentence and evaluate it as true. However, in a conversation in which 'he' refers to Aristotle, the sentence would be evaluated as false. The sentence cannot, however, be evaluated without a referent specified. Analogously, the unbound variables of QL1 formulas don't have any context-independent 'referent' (i.e., assignment). Therefore non-sentence formulas of QL1 are not to have determinate truth values on the models of QL1.

3.2.1 Models in QL1

An SL model for ϕ assigns a truth value to each sentence letter in ϕ, and that's it. QL1 has predicate letters and constants, which will require different kinds of assignments. Furthermore, QL1 has quantifiers. The quantifiers roughly correspond to English words such as *all* or *some*, so each model must specify a set of objects that the quantifiers range over. In other words, a QL1 model must fix a domain of objects over which to quantify. This is sometimes called a 'universe of discourse'.

Definition 3.20. A *model* for ϕ, $\mathfrak{m}$, consists of:

(1) an assignment of a truth value `T` or `F` to each sentence letter in ϕ;

(2) a single, non-empty set U, called the *universe* or *domain*;

(3) an assignment of a subset of U to each 1-place predicate in ϕ;

(4) an assignment of an element from U to each individual constant in ϕ.

We use the following notational conventions:

(1) Given some sentence letter, like P, $\mathfrak{m}(P)$ is the truth value $\mathfrak{m}$ assigns to P.

(2) $\mathfrak{m}(U)$ is the set $\mathfrak{m}$ assigns to U.

(3) Given a 1-place predicate, like G', $\mathfrak{m}(G')$ is the subset of U assigned to G' by $\mathfrak{m}$.

(4) Given an individual constant, like a, $\mathfrak{m}$(a) is the element from U assigned by $\mathfrak{m}$ to a.[7]

Earlier we said that the individual constants are roughly similar to proper names in English. One difference is that, in QL1, each individual constant in ϕ corresponds to exactly one object in the domain. In English, on the other hand, some proper names—e.g., 'John Smith'—correspond to more than one person, and some—e.g., 'Mordecai Alonzo Frazzle III'—do not correspond to any person. While each constant in ϕ is assigned an object from the domain, it is not required that different constants be assigned different objects.

We distinguish different models by affixing integers as subscripts to the symbol '$\mathfrak{m}$'. So, for example, $\mathfrak{m}_1$, $\mathfrak{m}_2$, $\mathfrak{m}_3$, ..., $\mathfrak{m}_{316}$, etc., are each different models.

As with SL, we have QL1 models for sets of sentences:

Definition 3.21. $\mathfrak{m}$ is a *model for a set of QL1 sentences* Δ iff $\mathfrak{m}$ is a model for each sentence in Δ.

There are also models that make assignments to all the sentence letters, constants, and 1-place predicates of QL1. Any such model is a model for every QL1 sentence. Let's call these *models for QL1*:

Definition 3.22. $\mathfrak{m}$ is a *model for QL1* iff $\mathfrak{m}$ is a model for every sentence of QL1.

We define truth in QL1 so that every model for ϕ fixes a unique truth value for ϕ. But there is a price we must pay for QL1's superior models. QL1 is more complicated than SL, so its definition of truth requires some additional metalinguistic tools. We must first define 'terms' and 'model variants'.

Definition 3.23. The *individual terms* of QL1 are the constants and variables of QL1.

We typically use 'q','r', 's', and 't' (along with subscripts) as MathEnglish metavariables for terms. This means that italic Roman 'q', 'r', 's', 't', and the Greek 'α', 'β', etc., can all be MathEnglish variables for QL variables. However, while 'α', 'β', etc. stand *only* for variables, the italic Roman letters can also range over QL1 constants. The use of metavariables for individual terms in the object language simplifies our notation considerably.

At this point we could define truth for sentences without quantifiers, though for now we offer only a short sketch. The sentence Ga is true on $\mathfrak{m}$ iff the element $\mathfrak{m}$ assigns to 'a' is in the set $\mathfrak{m}$ assigns to 'G'; i.e., iff $\mathfrak{m}$(a) $\in$ $\mathfrak{m}$(G). If the element $\mathfrak{m}$ assigns to 'a' isn't in the set assigned to 'G', Ga is false on $\mathfrak{m}$.

[7] We pause here to make two points for those keeping careful score: (1) We can think of models as functions from the set of basic symbols of QL1 (less the logical operators, variables, and parentheses) to the kinds of objects mentioned in definition 3.2 (on the previous page) (objects or subsets of U). (2) Those trying to keep careful track of the use/mention distinction should note that here we've been especially loose. We justify our laxity on the grounds that strict adherence to the distinction would clutter up our notation with confusing layers of quotes.

Quantifiers require more complexity. A sentence like $\forall xEx$ is true iff every element in the domain, U, is an element of $\mathfrak{m}(E)$. So if $\mathfrak{m}$ assigns U the set of even integers and $\mathfrak{m}(E) = U$, $\forall xEx$ is true. But if instead $\mathfrak{m}(E) = \{2, 4, 6\}$, then there are objects in the domain (e.g. 8) not in $\mathfrak{m}(E)$, and so $\forall xEx$ is false. For simple quantified sentences this quick definition is good enough; but it won't work for more complex sentences, such as $\forall x\exists y(Hy \rightarrow Gx)$.

To define truth for quantified sentences more precisely, we need a convenient notation for two models that make identical assignments everywhere except at one constant.

Definition 3.24. Let $\mathfrak{m}$ and $\mathfrak{m}^t$ be QL1 models and t be a term that is given some assignment by $\mathfrak{m}^t$. Then $\mathfrak{m}^t$ is a *t-variant* of $\mathfrak{m}$ iff for every term r such that $r \neq t$, $\mathfrak{m}(r) = \mathfrak{m}^t(r)$.

Put another way, a t-variant of $\mathfrak{m}$ is a model that makes all the same assignments as $\mathfrak{m}$ except possibly at constant t. One result of this definition is that any model $\mathfrak{m}$ that assigns something to t is a t-variant of itself. But what if $\mathfrak{m}$ doesn't assign anything to t? In that case, the t-variants are all the models that are identical with $\mathfrak{m}$ except with an additional assignment to t.

We generally denote t-variants of a model $\mathfrak{m}$ by affixing t as a superscript to '$\mathfrak{m}$'. For example, $\mathfrak{m}^{\mathrm{c}}$ is a c-variant of $\mathfrak{m}$. Then $\mathfrak{m}^{\mathrm{c}}$ and $\mathfrak{m}$ make all the same assignments, except possibly to the constant c. We extend this notation when the symbol denoting the original model is itself complex. So, given an c-variant of model $\mathfrak{m}$, i.e., $\mathfrak{m}^{\mathrm{c}}$, $\mathfrak{m}^{\mathrm{cd}}$ is a d-variant of $\mathfrak{m}^{\mathrm{c}}$. For another example, if $\mathfrak{m}_4^{\mathrm{e}}$ is an e-variant of $\mathfrak{m}_4$, then $\mathfrak{m}_4^{\mathrm{e}}$ and $\mathfrak{m}_4$ make identical assignments to everything except maybe e.

We need one more piece of notation before we get to the definition of truth.

Definition 3.25. If ϕ is a QL1 formula and t and s are terms, then $\phi s/t$ is the formula you get by replacing each unbound token of t in ϕ with a token of s.

Example 3.26.

(1) If ϕ is A, then $\phi y/x$ is A.

(2) If ϕ is Bx, then $\phi y/x$ is By.

(3) If ϕ is By, then $\phi y/x$ is By.

(4) If ϕ is Bx, then $\phi y/w$ is Bx.

(5) If ϕ is $\forall xBx$, then $\phi y/x$ is $\forall xBx$.

(6) If ϕ is $Cx \wedge \forall xBx$, then $\phi y/x$ is $Cy \wedge \forall xBx$.

(7) If ϕ is $\exists y(Cx \wedge \forall xBx)$, then $\phi y/x$ is $\exists y(Cy \wedge \forall xBx)$.

(8) If ϕ is $\exists y(Cx \wedge \forall xBx)$, then $\phi \mathrm{a}/x$ is $\exists y(C\mathrm{a} \wedge \forall xBx)$.

(9) If ϕ is $\exists y(Cx \wedge Bx)$, then $\phi \mathrm{a}/x$ is $\exists y(C\mathrm{a} \wedge B\mathrm{a})$.

3.2.2 Truth in a Model

We now define truth in a QL1 model.

Definition 3.27. The following clauses fix when a QL1 sentence θ is *true* (or *false*) on a model for θ, $\mathfrak{m}$:

(1) A sentence letter ϕ is true on $\mathfrak{m}$ iff $\mathfrak{m}(\phi) = \mathsf{T}$.

(2) An atomic sentence Pt with a 1-place predicate P and an individual term t is true on $\mathfrak{m}$ iff $\mathfrak{m}(t) \in \mathfrak{m}(P)$.

(3) A negation $\sim\phi$ is true on $\mathfrak{m}$ iff ϕ is false on $\mathfrak{m}$.

(4) A conjunction $(\phi_1 \wedge ... \wedge \phi_n)$ is true on $\mathfrak{m}$ iff all of $\phi_1, ..., \phi_n$ are true on $\mathfrak{m}$.

(5) A disjunction $(\phi_1 \vee ... \vee \phi_n)$ is true on $\mathfrak{m}$ iff at least one of $\phi_1, ..., \phi_n$ is true on $\mathfrak{m}$.

(6) A conditional $(\psi \rightarrow \phi)$ is true on $\mathfrak{m}$ iff the LHS ψ is false or the RHS ϕ is true on $\mathfrak{m}$, or both.

(7) A biconditional $(\psi \leftrightarrow \phi)$ is true on $\mathfrak{m}$ iff ψ and ϕ have the same truth value on $\mathfrak{m}$.

(8) A universal quantification $\forall\alpha\phi$ is true on $\mathfrak{m}$ iff $\phi t/\alpha$ is true on *all* t-variants of $\mathfrak{m}$, where t is the first constant not in ϕ.

(9) An existential quantification $\exists\alpha\phi$ is true on $\mathfrak{m}$ iff $\phi t/\alpha$ is true on *some* t-variant of $\mathfrak{m}$, where t is the first constant not in ϕ.

(10) A sentence ϕ is false on $\mathfrak{m}$ iff ϕ is not true on $\mathfrak{m}$.

For the following examples consult the models *Pos Int* and *States*, given in figure 3.1 (on the next page).

Example 3.28. The sentence $\sim G$b is true on the model *Pos Int*.

Proof: The model *Pos Int* assigns 9 to b, and the set of multiples of 7 to G. The number 9 is not a multiple of 7, so *Pos Int*(b) $\notin$ *Pos Int*(G). So, by the definition of truth, Gb is false on *Pos Int*. Therefore, by the definition of truth for $\sim$, *Pos Int* makes $\sim G$b true.

■

> *Pos Int*(b) $\notin$ *Pos Int*(G) asserts that what *Pos Int* assigns to b, 9, is not an element of the set that *Pos Int* assigns to G. To check this, we need only to look at the set assigned to G and see whether 7 is a member; it's not, since 9 is not a multiple of 7.

Example 3.29. The sentence $(G\text{d} \rightarrow D\text{e})$ is true on the model *Pos Int* (3.1).

	Symbol	**Model**	
		Pos Int	**States**
Universe:		The set of positive integers	The set of US states (2024)
Sent. Let.:	A	true	false
	B	true	false
	C	false	true
	D	true	false
	E	true	false
	G	false	true
Constants:	a	1	Louisiana
	b	9	Maine
	c	72	Georgia
	d	3	Nebraska
	e	1	New Mexico
	f	2	Texas
1-place:	A'	all pos int	Midwestern
	B'	empty set	name with > 5 letters
	C'	even	Coastal
	D'	odd	on the Pacific coast
	E'	prime	{Ohio}
	G'	multiple of 7	{Ohio,Alabama}

Figure 3.1: Two QL1 models

Proof: $(G\mathrm{d} \rightarrow D\mathrm{e})$ is true on *Pos Int* iff the LHS is false or the RHS is true. *Pos Int* assigns the number 3 to d and the set of multiples of 7 to G. But 3 isn't a multiple of 7, so *Pos Int*(d) $\notin$ *Pos Int*(G). It follows that $G\mathrm{d}$ is false on *Pos Int*. Therefore $(G\mathrm{d} \rightarrow D\mathrm{e})$ is true on *Pos Int*. ■

Example 3.30. The sentence $A\mathrm{a} \vee G\mathrm{c}$ is false on the model *States* (3.1).

Proof: The model *States* assigns Louisiana to a, Georgia to c, the set of Midwestern states to A, and the set {Ohio, Alabama} to G.

$A\mathrm{a}$ is true on *States* iff *States*(a) $\in$ *States*(A). But Louisiana isn't a Midwestern state. So $A\mathrm{a}$ is false on *States*. $G\mathrm{c}$ is true on *States* iff *States*(c) $\in$ *States*(G). But *States* assigns {Ohio, Alabama} to G. Georgia isn't in that set, so *States* makes $G\mathrm{c}$ false.

$A\mathrm{a}$ and $G\mathrm{c}$ are false on *States*, so *States* makes $A\mathrm{a} \vee G\mathrm{c}$ false. ■

Pay attention to the full structure of a sentence when assessing its truth value. The sentences $\forall xDx \rightarrow \forall xGx$ and $\forall x(Dx \rightarrow Gx)$ may look similar, but they mean very different things. Note that the first has '$\rightarrow$' as its main connective, whereas the main connective of the second is '$\forall$'. We show that their truth values come apart on *Pos Int* in the next two examples.

Example 3.31. $\forall xDx \rightarrow \forall xGx$ is true on the model *Pos Int* (3.1).

Proof: $\forall xDx \rightarrow \forall xGx$ is true on *Pos Int* iff *Pos Int* either makes the LHS is false or the RHS true. By the definition of truth for $\forall$, $\forall xDx$ is true on *Pos Int* iff $D\mathrm{a}$ is true on all a-variants of *Pos Int*.

> The definition of truth for $\forall$ has us substitute the first constant not in Dx. Since it has no constant we use the first one: a. The result of the substitution is $D\mathrm{a}$.

Pos Int assigns the set of odd numbers to D. Consider the a-variant of *Pos Int*, *Pos Int*$^{\mathrm{a}}$, such that *Pos Int*$^{\mathrm{a}}$(a) is 2. Clearly 2 is not an odd number. It follows that $D\mathrm{a}$ is false on *Pos Int*$^{\mathrm{a}}$, and so $\forall xDx$ is false on *Pos Int*. Therefore $\forall xDx \rightarrow \forall xGx$ is true on *Pos Int*. ■

Example 3.32. $\forall x(Dx \rightarrow Gx)$ is false on *Pos Int* (3.1).

Proof: By the definition of truth for $\forall$, $\forall x(Dx \rightarrow Gx)$ is true on *Pos Int* iff every a-variant of *Pos Int* makes $D\mathrm{a} \rightarrow G\mathrm{a}$ true. *Pos Int* assigns the set of odd numbers to D and the set of multiples of 7 to G. Let there be an a-variant of *Pos Int*, *Pos Int*$^{\mathrm{a}}$, such that *Pos Int*$^{\mathrm{a}}$(a) = 3. The number 3 is odd and not a multiple of 7. So *Pos Int*$^{\mathrm{a}}$(a) $\in$ *Pos Int*$^{\mathrm{a}}$(D) and *Pos Int*$^{\mathrm{a}}$(a) $\notin$ *Pos Int*$^{\mathrm{a}}$(G). Thus, *Pos Int*$^{\mathrm{a}}$ makes $D\mathrm{a}$ true and $G\mathrm{a}$ false, which in turn makes $D\mathrm{a} \rightarrow G\mathrm{a}$ false. Therefore $\forall x(Dx \rightarrow Gx)$ is false on *Pos Int*. ■

Let's compare another pair of superficially similar sentences, this time with existential quantifiers: $\exists x(Cx \wedge Dx)$ and $(\exists xCx \wedge \exists xDx)$.

Example 3.33. $\exists x(Cx \wedge Dx)$ is false on *Pos Int* (3.1).

Proof: By the definition of truth for $\exists$, $\exists x(Cx \wedge Dx)$ is true on *Pos Int* iff there is an a-variant of *Pos Int* that makes Ca $\wedge$ Da true. *Pos Int* assigns the even numbers to C and the odd numbers to D. There is no element in the domain of *Pos Int* that is both odd and even. So there is no a-variant of *Pos Int* that makes both Ca and Da true. Hence Ca $\wedge$ Da is false on all a-variants. Therefore $\exists x(Cx \wedge Dx)$ is false on *Pos Int*. ■

Example 3.34. $(\exists xCx \wedge \exists xDx)$ is true on *Pos Int* (3.1).

Proof: $(\exists xCx \wedge \exists xDx)$ is true on *Pos Int* iff $\exists xCx$ and $\exists xDx$ are true on *Pos Int*.

By the definition of truth for $\exists$, $\exists xCx$ is true on *Pos Int* iff there is an a-variant of *Pos Int* that makes Ca true. *Pos Int* assigns the even numbers to C. Let *Pos Int*$^{\text{a}}$ be an a-variant that assigns 4 to a. The number 4 is even, so *Pos Int*$^{\text{a}}$(a) $\in$ *Pos Int*$^{\text{a}}$(C). Thus, *Pos Int*$^{\text{a}}$ makes Ca true, and so *Pos Int* makes $\exists xCx$ true.

By the definition of truth for $\exists$, $\exists xDx$ is true on *Pos Int* iff there is an a-variant of *Pos Int* that makes Da true. *Pos Int* assigns the odd numbers to D. Let *Pos Int*$^{\text{a}}$ be an a-variant that assigns 3 to a. The number 3 is odd, so *Pos Int*$^{\text{a}}$(a) $\in$ *Pos Int*$^{\text{a}}$(D). Thus, *Pos Int*$^{\text{a}}$ makes Da true. It follows that *Pos Int* makes $\exists xDx$ true.

Therefore *Pos Int* makes $(\exists xCx \wedge \exists xDx)$ true. ■

On model *Pos Int* we can interpret $\exists x(Cx \wedge Dx)$ as saying, roughly that there is some positive integer that is both even and odd. That's clearly false. By contrast, $(\exists xCx \wedge \exists xDx)$ means, roughly, that there is some even positive integer and there is some odd positive integer, which is true.

We can calculate the truth value of a QL1 sentence ϕ on some $\mathfrak{m}$ iff $\mathfrak{m}$ is a model for ϕ. If $\mathfrak{m}$ *isn't* a model for ϕ, then ϕ has no truth value on it. Be careful: if $\mathfrak{m}_1$ is a model for ϕ and $\mathfrak{m}_2$ is a model for ψ, it doesn't follow that either $\mathfrak{m}_1$ or $\mathfrak{m}_2$ is a model for, say, $\phi \rightarrow \psi$.

3.2.3 Minimal Models in QL1

For the examples above we used models with assignments that are not referenced in the sentences we evaluated. The Minimal Model theorem below proves that assignments to irrelevant predicates, constants, etc. don't matter for the truth value of a sentence.

Definition 3.35. Model $\mathfrak{m}$ is a *minimal model for* ϕ iff $\mathfrak{m}$ makes the minimum assignments necessary for $\mathfrak{m}$ to be a model for ϕ.

That is, a minimal model makes assignments to the universe U, to each sentence letter, constant, and 1-place predicate in ϕ, but to nothing else. Consider the sentence $(\exists xCx \wedge \exists xDx)$. This sentence has two 1-place predicates, C and D, and no sentence

letters or constants. A minimal model for this sentence makes no assignment to any sentence letter or constant. It assigns subsets of U to C and D, but makes no assignment to any other 1-place predicate. Most logic texts do not define minimal models but implicitly use them. We prefer to define them explicitly.

When calculating the value of a sentence ϕ on a model, we only need to worry about the assignments to the symbols in ϕ and the universe. We can ignore the other assignments. The following theorem demonstrates this. This is the QL1 version of theorem 2.70 in chapter 2.

Theorem 3.36. Let ϕ be any QL1 sentence. If there are two models for ϕ, $\mathfrak{m}_1$ and $\mathfrak{m}_2$, that have the same domain, U, and make the same assignments for all the sentence letters, individual constants, and 1-place predicates contained in ϕ, then ϕ is true on $\mathfrak{m}_1$ iff ϕ is true on $\mathfrak{m}_2$.

Proof:

Base Step: Let θ be a sentence of order 1. θ is either (i) a sentence letter, or (ii), a 1-place predicate followed by a constant.

(i) θ is a sentence letter. We assume that $\mathfrak{m}_1$ and $\mathfrak{m}_2$ make the same assignments to all the sentence letters. Then θ is true on $\mathfrak{m}_1$ iff θ is true on $\mathfrak{m}_2$.

(ii) $\theta = Pt$, where P is a 1-place predicate and t is a constant. We assume that $\mathfrak{m}_1$ and $\mathfrak{m}_2$ make the same assignments to all the constants and 1-place predicates. Then $\mathfrak{m}_1(P) = \mathfrak{m}_2(P)$ and $\mathfrak{m}_1(t) = \mathfrak{m}_2(t)$. It follows that $\mathfrak{m}_1(t) \in \mathfrak{m}_1(P)$ iff $\mathfrak{m}_2(t) \in \mathfrak{m}_2(P)$. Thus, θ is true on $\mathfrak{m}_1$ iff θ is true on $\mathfrak{m}_2$.

Inheritance Step:

Recursive Assumption: Assume that θ is a QL1 sentence of order n. Assume also that $\mathfrak{m}_1$ and $\mathfrak{m}_2$ are any two models for θ that make the same assignments to all the sentence letters, individual constants, and 1-place predicates contained in θ. Then let θ be true on $\mathfrak{m}_1$ iff θ is true on $\mathfrak{m}_2$.

Negation, Conditional, Biconditional, Disjunction, Conjunction: The reasoning for these is the same as in the corresponding clauses of theorem 2.70 in chapter 2.

Quantifier Preface: Let $\forall\alpha\theta$ and $\exists\alpha\theta$ be sentences of order $n+1$. By RA, $\mathfrak{m}_1$ and $\mathfrak{m}_2$ have the same universe, U. It follows that for each element $u \in U$, there is a t-variant of $\mathfrak{m}_1$, $\mathfrak{m}_1^t$, and a t-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^t$, such that $\mathfrak{m}_1^t(t) = \mathfrak{m}_2^t(t) = u$. Since $\mathfrak{m}_1$ is a model for $\forall\alpha\theta$ and $\exists\alpha\theta$, each $\mathfrak{m}_1^t$ is a model for $\theta t/\alpha$. By similar reasoning each $\mathfrak{m}_2^t$ is a model for $\theta t/\alpha$. $\theta t/\alpha$ is of order n. So by RA, for each pair of t-variants such that $\mathfrak{m}_1^t(t) = \mathfrak{m}_2^t(t)$, $\theta t/\alpha$ is true on $\mathfrak{m}_1^t$ iff it's true on $\mathfrak{m}_2^t$.

Universal Quantification: ($\Rightarrow$) Let $\forall\alpha\theta$ be true on $\mathfrak{m}_1$. Then, by the definition of truth of $\forall$, all t-variants of $\mathfrak{m}_1$ make $\theta t/\alpha$ true. Since by the quantifier preface there is a corresponding t-variant of $\mathfrak{m}_2$ for each t-variant of $\mathfrak{m}_1$,

it follows that all t-variants of $\mathfrak{m}_2$ make $\theta t/\alpha$ true. So, by the definition of truth for $\forall$, $\forall\alpha\theta$ is true on $\mathfrak{m}_2$.
($\Leftarrow$) Let $\forall\alpha\theta$ be true on $\mathfrak{m}_2$. Then, by the definition of truth of $\forall$, all t-variants of $\mathfrak{m}_2$ make $\theta t/\alpha$ true. Since by the quantifier preface there is a corresponding t-variant of $\mathfrak{m}_1$ for each t-variant of $\mathfrak{m}_2$, it follows that all t-variants of $\mathfrak{m}_1$ make $\theta t/\alpha$ true. So, by the definition of truth for $\forall$, $\forall\alpha\theta$ is true on $\mathfrak{m}_1$.
Thus $\forall\alpha\theta$ is true on $\mathfrak{m}_1$ iff $\forall\alpha\theta$ is true on $\mathfrak{m}_2$.

Existential Quantification: ($\Rightarrow$) Let $\exists\alpha\theta$ be true on $\mathfrak{m}_1$. Then, by the definition of truth of $\exists$, there is some t-variant of $\mathfrak{m}_1$ that makes $\theta t/\alpha$ true. Since by the quantifier preface there is a corresponding t-variant of $\mathfrak{m}_2$ for each t-variant of $\mathfrak{m}_1$, it follows that there is some t-variant of $\mathfrak{m}_2$ that makes $\theta t/\alpha$ true. So, by the definition of truth for $\exists$, $\exists\alpha\theta$ is true on $\mathfrak{m}_2$.
($\Leftarrow$) Let $\exists\alpha\theta$ be true on $\mathfrak{m}_2$. Then, by the definition of truth of $\exists$, there is some t-variant of $\mathfrak{m}_2$ that makes $\theta t/\alpha$ true. Since by the quantifier preface there is a corresponding t-variant of $\mathfrak{m}_1$ for each t-variant of $\mathfrak{m}_2$, it follows that there is some t-variant of $\mathfrak{m}_1$ that makes $\theta t/\alpha$ true. So, by the definition of truth for $\exists$, $\exists\alpha\theta$ is true on $\mathfrak{m}_1$.
Thus $\exists\alpha\theta$ is true on $\mathfrak{m}_1$ iff $\exists\alpha\theta$ is true on $\mathfrak{m}_2$.

Closure Step: There is no other way to form a QL1 sentence ϕ, so the above clauses are sufficient to show that if $\mathfrak{m}_1$ and $\mathfrak{m}_2$ have the same domain and make the same assignments, then ϕ is true on $\mathfrak{m}_1$ iff ϕ is true on $\mathfrak{m}_2$.

■

3.2.4 Logical Truth: QT, QF, & QC

Just as we have the notions of *logical truth*, *falsity*, and *contingency* for SL (see section 2.2.3), we have analogous notions for QL1.

Definition 3.37. A sentence ϕ of QL is *quantificationally true* (QT) iff it is true on every model for ϕ.

Definition 3.38. A sentence ϕ of QL is *quantificationally false* (QF) iff it is false on every model for ϕ.

Definition 3.39. A sentence ϕ of QL is *quantificationally contingent* (QC) iff there's at least one model $\mathfrak{m}_1$ on which it's true and at least one model $\mathfrak{m}_2$ on which it's false.

Example 3.40. Prove that $\forall x(Bx \vee \sim Bx)$ is QT.

Proof: $\forall x(Bx \vee \sim Bx)$ is true on $\mathfrak{m}$ iff every a-variant of $\mathfrak{m}$, $\mathfrak{m}^{\mathrm{a}}$, makes $B\mathrm{a} \vee \sim B\mathrm{a}$ true. One of two things must be true of each $\mathfrak{m}^{\mathrm{a}}$. Either $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}^{\mathrm{a}}(B)$ or $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \notin \mathfrak{m}^{\mathrm{a}}(B)$.

If $\mathfrak{m}^{a}(a) \in \mathfrak{m}^{a}(B)$, then Ba is true on $\mathfrak{m}^{a}$, and so is Ba $\vee \sim B$a. If $\mathfrak{m}^{a}(a) \notin \mathfrak{m}^{a}(B)$, then $\sim B$a is true on $\mathfrak{m}^{a}$, and so is Ba $\vee \sim B$a. So, Ba $\vee \sim B$a is true on every $\mathfrak{m}^{a}$. It follows that $\forall x(Bx \vee \sim Bx)$ is true on $\mathfrak{m}$. Nothing particular was assumed about $\mathfrak{m}$, so $\forall x(Bx \vee \sim Bx)$ is true on all models. Therefore, $\forall x(Bx \vee \sim Bx)$ is QT. ■

Example 3.41. Prove that $\forall x(Bx \wedge \sim Bx)$ is QF.

Proof: $\forall x(Bx \wedge \sim Bx)$ is true on $\mathfrak{m}$ iff every a-variant of $\mathfrak{m}$, $\mathfrak{m}^{a}$, makes Ba $\wedge \sim B$a true. There is some a-variant $\mathfrak{m}^{a}$ such that either $\mathfrak{m}^{a}(a) \in \mathfrak{m}^{a}(B)$ or $\mathfrak{m}^{a}(a) \notin \mathfrak{m}^{a}(B)$. If $\mathfrak{m}^{a}(a) \in \mathfrak{m}^{a}(B)$ then $\sim B$a is false on $\mathfrak{m}^{a}$, and so is Ba$\wedge\sim B$a. Otherwise $\mathfrak{m}^{a}(a) \notin \mathfrak{m}^{a}(B)$, in which case Ba is false on $\mathfrak{m}^{a}$, and so is Ba $\wedge \sim B$a. In both cases it follows that $\forall x(Bx \wedge \sim Bx)$ is false on $\mathfrak{m}$. Nothing particular was assumed about $\mathfrak{m}$, so $\forall x(Bx \wedge \sim Bx)$ is false on all models. Therefore, $\forall x(Bx \wedge \sim Bx)$ is QF. ■

In the last two examples we give proofs that the sentences in question are either QT or QF. To show that a QL1 sentence ϕ is QC, we use a different strategy: provide one model on which ϕ is true and another on which ϕ is false.

For example, we determined earlier that $\forall xDx \rightarrow \forall xGx$ is true on the model *Pos Int*. To show that it's QC we must provide another model in which it's false. We can modify *Pos Int* to make a new model, *Pos Int**. Let *Pos Int** assign the entire domain to D so that it makes $\forall xDx$ true. We can keep the assignment *Pos Int* makes to G, the multiples of 7. So, on *Pos Int**, $\forall xGx$ is false. Thus, $\forall xDx \rightarrow \forall xGx$ is false on *Pos Int**.

The definition of model only requires that predicates be assigned a subset of the domain. The assigned subset can be the entire domain or the empty set. You will find that using the empty set or the entire domain as assignments is often helpful in constructing models for a desired outcome.

Example 3.42. The sentence $\forall xBx \vee \forall x\sim Bx$ is QC.

Proof: Let $\mathfrak{m}_1$ be a model such that the universe $U = \{1\}$ and $\mathfrak{m}_1(B) = \{1\}$. $\mathfrak{m}_1$ makes $\forall xBx$ true, which in turn makes $\forall xBx \vee \forall x\sim Bx$ true.

> When we remove the quantifier from $\forall xBx$ and replace the variable with a constant, the result is Ba. Everything in the universe is also in the set assigned to B. So, no matter what an a-variant assigns to the constant, the result is true.

Let $\mathfrak{m}_2$ be a model such that the universe $U = \{1, 2\}$ and $\mathfrak{m}_2(B) = \{1\}$. $\forall xBx$ is false on $\mathfrak{m}_2$ because not everything in the domain is in the set assigned to B. And $\forall x\sim Bx$ is false on $\mathfrak{m}_2$ because the model doesn't make B the empty set. Then $\forall xBx \vee \forall x\sim Bx$ is false in $\mathfrak{m}_2$, because both disjuncts are false. Therefore, the sentence $\forall xBx \vee \forall x\sim Bx$ is QC. ■

Example 3.43. Prove that $\forall xDx \rightarrow \sim\exists x\sim Dx$ is QT.

Proof: Assume for indirect proof there is some model $\mathfrak{m}$ that makes $\forall x Dx \rightarrow \sim\exists x \sim Dx$ false. Thus, $\forall x Dx$ is true on $\mathfrak{m}$ and $\sim\exists x \sim Dx$ is false on $\mathfrak{m}$. From the truth of $\forall x Dx$ it follows that every a-variant of $\mathfrak{m}$ makes Da true. Because $\mathfrak{m}$ makes $\sim\exists x \sim Dx$ false, it follows that $\exists x \sim Dx$ is true. Given that $\mathfrak{m}$ makes $\exists x \sim Dx$ true, there must be some a-variant, $\mathfrak{m}^{\text{a}}$, that makes $\sim D$a true. On $\mathfrak{m}^{\text{a}}$, Da must be false. But it was already shown that every a-variant of $\mathfrak{m}$ makes Da true. Thus there is no model such that $\forall x Dx \rightarrow \sim\exists x \sim Dx$ is false. It is therefore true on all models, and so is QT. ■

3.3 Entailment and other Relations

We defined the following terms in SL: entailment, equivalence, contradictory, contrary, subcontrary, and logical independence. Now we define these for QL1 sentences. While the SL definitions refer to models of SL sentences, their QL1 counterparts refer to models of QL1 sentences. To mark this difference, we talk of *truth functional* entailment, *truth functional* equivalence, etc., for SL sentences, and *quantificational* entailment, *quantificational* equivalence, etc., for QL1. We don't want to overstate the difference; the concepts underlying the definitions are essentially the same.

Definition 3.44. A set Δ of QL1 sentences *quantificationally entails* a QL1 sentence θ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Put another way, Δ entails θ iff every model that makes all sentences in Δ true also makes θ true. As we did with SL (section 2.3.1), we give two narrower consequences of this definition.

(1) A finite set of QL1 sentences $\phi_1, \dots, \phi_n$ quantificationally entails another QL1 sentence θ iff every model for $\phi_1, \dots, \phi_n$, and θ either makes at least one of $\phi_1, \dots, \phi_n$ false or makes θ true.

(2) A sentence ϕ of QL quantificationally entails another sentence θ of QL1 iff every model for ϕ and θ either makes ϕ false or makes θ true.

We continue to use the double turnstile to represent the entailment relation. So if ϕ entails θ, we write '$\phi \models \theta$'. If the sentences $\phi_1, \dots, \phi_n$ entail θ, we write '$\phi_1, \dots, \phi_n \models \theta$'. And if a set Δ of sentences entails θ we write '$\Delta \models \theta$'.

Example 3.45. Show whether the following holds: $\forall x Gx \models G$a.

Proof: Assume a model $\mathfrak{m}$ such that $\forall x Gx$ is true. By the definition of truth for $\forall$, it follows that Ga is true on all a-variants of $\mathfrak{m}$. The model $\mathfrak{m}$ is an a-variant of itself, so Ga is true on $\mathfrak{m}$. Nothing was assumed about the assignments $\mathfrak{m}$ makes, so for each model on which the LHS is true, the RHS is also true. Therefore $\forall x Gx \models G$a. ■

Example 3.46. Show that $\forall x Gx \models G$b.

This entailment is slightly harder to prove than the last. It's a quirk of our definition of truth that makes the last so easy to establish. By changing the constant in the sentence on the RHS from 'a' to 'b', we add a few steps to our proof.

Proof: Assume a model $\mathfrak{m}$ such that $\forall x Gx$ is true. Then Ga is true on all a-variants of $\mathfrak{m}$. All a-variants of $\mathfrak{m}$ have the same universe as $\mathfrak{m}$. So there is some a-variant, $\mathfrak{m}^{\mathrm{a}}$, such that $\mathfrak{m}^{\mathrm{a}}(a) = \mathfrak{m}(\mathrm{b})$. It follows that $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}^{\mathrm{a}}(G)$. $\mathfrak{m}^{\mathrm{a}}$ and $\mathfrak{m}$ assign the same set to G, so $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}(G)$. And because $\mathfrak{m}^{\mathrm{a}}(\mathrm{a})$ is the same element as $\mathfrak{m}(\mathrm{b})$, it follows that $\mathfrak{m}(\mathrm{b}) \in \mathfrak{m}(G)$. And so Gb is true on $\mathfrak{m}$. Nothing was assumed about the assignments $\mathfrak{m}$ makes, so for each model on which the LHS is true, the RHS is also true. Therefore $\forall x Gx \models G\mathrm{b}$. ■

Example 3.47. Show whether $\forall x(Cx \to Dx), C\mathrm{o} \models D\mathrm{o}$.

Proof: The entailment holds. Assume some model $\mathfrak{m}$ such that $\forall x(Cx \to Dx)$ and Co are true. By the definition of truth for $\forall$, it follows that $(C\mathrm{a} \to D\mathrm{a})$ is true on all a-variants of $\mathfrak{m}$. All a-variants of $\mathfrak{m}$ have the same universe as $\mathfrak{m}$. So there is an a-variant such that $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) = \mathfrak{m}(\mathrm{o})$.

The model $\mathfrak{m}$ makes Co true, so $\mathfrak{m}(\mathrm{o}) \in \mathfrak{m}(C)$. Because $\mathfrak{m}(\mathrm{o}) = \mathfrak{m}^{\mathrm{a}}(\mathrm{a})$ and $\mathfrak{m}^{\mathrm{a}}(C) = \mathfrak{m}(C)$, it follows by substitution that $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}^{\mathrm{a}}(C)$. Hence, $\mathfrak{m}^{\mathrm{a}}$ makes Ca true. And because $(C\mathrm{a} \to D\mathrm{a})$ is true on $\mathfrak{m}^{\mathrm{a}}$, it follows that $\mathfrak{m}^{\mathrm{a}}$ makes Da true. From this it follows that $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}^{\mathrm{a}}(D)$. $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) = \mathfrak{m}(\mathrm{o})$ and $\mathfrak{m}^{\mathrm{a}}(D) = \mathfrak{m}(D)$, so by substitution we get: $\mathfrak{m}(\mathrm{o}) \in \mathfrak{m}(D)$. Thus, $\mathfrak{m}$ makes Do true.

Any model that makes the LHS true also makes the RHS true. Therefore the entailment holds. ■

This entailment resembles the argument discussed at the beginning of the chapter: (1) All women are mortal, (2) Ophelia is a woman, therefore (3) Ophelia is mortal. To see this, interpret o as Ophelia, C as the set of women, and D as the set of mortals.

Consider an entailment $\Delta \models \phi$ that holds when the set Δ is empty, i.e. such that $\models \phi$. By the definition of $\models$ every model must either make a sentence on the left false or the sentence on the right true. But in this case there are no sentences on the LHS. So every model must make the RHS, ϕ, true. Therefore, as was the case with SL in chapter 2, $\models \phi$ iff ϕ is QT.

Definition 3.48. Two QL1 sentences θ and ϕ are *quantificationally equivalent* iff all models for θ and ϕ assign them the same truth value.

Definition 3.49. Two QL1 sentences θ and ϕ are *quantificationally contradictory* iff all models for θ and ϕ assign them opposite truth values.

Definition 3.50. Two QL1 sentences θ and ϕ are *quantificationally contrary* iff they cannot both be true in the same model $\mathfrak{m}$.

Definition 3.51. Two QL1 sentences θ and ϕ are *quantificationally subcontrary* iff they cannot both be false in the same model $\mathfrak{m}$.

Definition 3.52. Two QL1 sentences θ and ϕ are *quantificationally independent* iff there are four models:

(1) A model in which both θ and ϕ are true;

(2) A model in which both θ and ϕ are false;

(3) A model in which θ is true and ϕ is false; and

(4) A model in which θ is false and ϕ is true.

First, a minor note. These definitions only make sense for QL1 sentences. We do not assess formulas that *aren't* sentences for truth value, so none of the definitions above apply to them.

Except for the fact that these definitions apply to sentences of QL1 instead of SL, and models for QL1 sentences instead of models for SL sentences, they are the same as the corresponding ones for SL. We might say that these definitions have the same "structure". The *ideas* of equivalence, being contradictory, etc., haven't changed, even though the details of the definitions are slightly different.

The following four facts from SL also hold in QL1 (compare with the examples in section 2.3.4):

(1) Contradictory sentences are also contrary, but sentences can be contrary without being contradictory: e.g. $C \wedge D$ and $C \wedge \sim D$.

(2) Contradictory sentences are also subcontrary, but sentences can be subcontrary without being contradictory: e.g. D and $C \vee \sim D$.

(3) If two sentences are both contrary and subcontrary, they are contradictory.

(4) Any two atomic sentences are independent of each other.

Because every sentence of SL is also a sentence of QL1 we can reuse the examples from the previous chapter.

Finally, recall that in SL we have theorem 2.47, on page 33: for all SL sentences ϕ and θ, $\phi \models \theta$ iff $\models (\phi \rightarrow \theta)$. The same theorem holds for QL1 sentences and the proof is essentially the same.

Theorem 3.53. QL1 Exportation Theorem: For all QL1 sentences ϕ and θ, $\phi \models \theta$ iff $\models (\phi \rightarrow \theta)$.

In addition, all the generalizations of this theorem (2.48) also hold for QL1 sentences, and again the proofs are essentially same.

3.4 Exercises

3.4.1 Formulas, Order, and Subformulas

Which of the following are QL1 *formulas*? For those that are formulas, what is their order? How many subformulas does each have?

1. $\forall x(H'x \rightarrow G'x)$
2. $\forall x(H'x \rightarrow G''x)$
3. $\forall x(H'x \rightarrow G'_7x)$
4. $\forall x \forall z(H'x \rightarrow G''xy)$
5. $\exists y \forall x(H'x \rightarrow G'x)$
6. $\forall t(H'x \rightarrow G'x)$
7. $H'x \vee G'x$
8. $\forall x(H'y \wedge G'z)$

	Symbol	**Model**	
		Pos Int	**States**
Universe:		The set of positive integers	The set of US states (2024)
Sent. Let.:	A	true	false
	B	true	false
	C	false	true
	D	true	false
	E	true	false
	G	false	true
Constants:	a	1	Louisiana
	b	9	Maine
	c	72	Georgia
	d	3	Nebraska
	e	1	New Mexico
	f	2	Texas
1-place:	A'	all pos int	Midwestern
	B'	empty set	name with > 5 letters
	C'	even	Coastal
	D'	odd	on the Pacific coast
	E'	prime	{Ohio}
	G'	multiple of 7	{Ohio,Alabama}

Table 3.2: Example Models

3.4.2 Truth in a Model

Give the truth value of each of the following sentences on both of the models found in table 3.2 (on the previous page).

1. $\exists x Gx$
2. $\sim\exists x Gx$
3. $\exists x \sim Gx$
4. $\forall x Gx$
5. $\sim\forall x Gx$
6. $\forall x \sim Gx$
7. $\exists x Cx \wedge \exists x Dx$
8. $\exists x(Cx \wedge Dx)$
9. $\sim\exists x(Cx \wedge Dx)$
10. $\forall x(Cx \wedge Dx)$
11. $\forall x(Cx \rightarrow Dx)$
12. $\forall x Cx \rightarrow \forall x Dx$
13. $\sim\forall x(Cx \rightarrow Dx)$
14. $\exists x(Cx \rightarrow Dx)$

3.4.3 Quantificational Truth Problems

For each sentence below, say whether it's a quantificational truth. If so, prove it. If not, give a model $\mathfrak{m}$ that makes it false.

1. $\forall y(Ay \rightarrow By) \vee \forall y(By \rightarrow Ay)$
2. $\exists y(Ay \rightarrow By) \vee \exists y(By \rightarrow Ay)$
3. $\forall y(Ay \rightarrow By) \rightarrow (\exists y Ay \rightarrow \exists y By)$
4. $\exists y(Ay \rightarrow By) \rightarrow (\exists y Ay \rightarrow \exists y By)$
5. $\exists y(Ay \rightarrow By) \rightarrow (\forall y Ay \rightarrow \forall y By)$
6. $\forall y \sim Ay \rightarrow \sim\exists y Ay$
7. $\sim\exists y Ay \rightarrow \forall y \sim Ay$
8. $\sim\forall y Ay \rightarrow \exists y \sim Ay$
9. $\forall y(Ay \rightarrow By) \rightarrow (\forall y Ay \rightarrow \forall y By)$
10. $\forall z(Az \rightarrow (Bz \vee Cz)) \rightarrow (\forall z(Az \rightarrow Bz) \vee \forall z(Az \rightarrow Cz))$
11. $\forall y(Ay \rightarrow By) \rightarrow (\forall y(By \rightarrow Cy) \rightarrow \forall y(Ay \rightarrow Cy))$
12. $\forall y(Ay \rightarrow By) \rightarrow (\forall y(Cy \rightarrow By) \rightarrow \forall y(Ay \rightarrow Cy))$
13. $\forall y(Ay \rightarrow By) \rightarrow (\exists y(By \rightarrow Cy) \rightarrow \forall y(Ay \rightarrow Cy))$
14. $\forall y(Ay \rightarrow By) \rightarrow (\exists y(By \rightarrow Cy) \rightarrow \exists y(Ay \rightarrow Cy))$

3.4.4 Entailment Problems for QL1

For each entailment below, either prove that it holds or show that it doesn't hold by giving a model that make the sentences on the LHS of the turnstile true and the sentence on the RHS false.

1. $\forall y(Ay \rightarrow By),\ \forall yAy \models \forall yBy$
2. $\forall y(Ay \rightarrow By),\ \exists yAy \models \exists yBy$
3. $\exists y(Ay \rightarrow By),\ \exists yAy \models \exists yBy$
4. $\forall yAy \rightarrow \forall yBy \models \forall y(Ay \rightarrow By)$
5. $\exists y(Ay \vee By) \models \exists yAy \vee \exists yBy$
6. $\exists y(Ay \rightarrow By),\ \forall yAy \models \forall yBy$
7. $\forall z(Az \rightarrow (Bz \vee Cz)) \models (\forall z(Az \rightarrow Bz) \vee \forall z(Az \rightarrow Cz))$
8. $\forall y(Ay \rightarrow By),\ \exists y(By \rightarrow Cy) \models \exists y(Ay \rightarrow Cy)$

3.4.5 Relations Between QL1 Sentences

For each sentence below, say whether it entails, is entailed by, is equivalent to, contradicts, is contrary to, is subcontrary to, or is independent from each of the other sentences.

1. $\forall z(Gz \rightarrow Dz)$
2. $\forall zGz \rightarrow \forall zDz$
3. $\exists z(Gz \wedge \sim Dz)$
4. $\exists z(Gz \wedge Dz)$
5. $\forall z(Gz \wedge Dz)$
6. $\exists z(Gz \rightarrow Dz)$
7. $\forall z(Gz \rightarrow \sim Dz)$

Chapter 4

Quantifier Language II

4.1 The Language QL

4.1.1 Symbols

In this chapter we add many-place predicates to QL1. The resulting language is QL, and its development was a significant event in the history of logic.[1] The 2-place predicates correspond roughly to what you get if you take an English sentence and remove two names, leaving blanks, e.g.:

1. 'Goliath is taller than David' ⇒ '_____ is taller than _____'
2. 'Juliet Capulet loves Romeo Montague' ⇒ '_____ loves _____'

We may think of 2-place predicates as representing a 2-place *relation*. The 'taller than' relation holds between two objects when one is taller than the other. The 'loves' relation holds when one person—or object of whatever kind—loves another. We may understand 3-place predicates in a similar way. For example:

3. 'Three is between two and four' ⇒ '_____ is between _____ and _____'

For any $n \geq 2$, an n-place predicate represents an n-place relation. The introduction of many-place predicates significantly increases the power of our formal language. Consider the following argument:

4. All horses are animals.

   Therefore,

5. All horses' tails are animals' tails.

[1] The development of QL goes back to Gottlob Frege (1879; 1966 [1891]; 1893/1903), O. H. Mitchell (1883) and Charles S. Peirce (1883), with Frege's work being independent of and unknown to the latter two. See Church 1956: 288 and Hodges 2001b: 34. It's probably safe to say that Frege and Peirce/Mitchell developed quantificational logic independently, but the extent to which Peirce and his students (like Mitchell) knew of Frege's work is a matter of debate. It's clear they at least knew of Frege. E.g., Ladd-Franklin (1883) cites Frege's (1879) through a review of it by Ernst Schröder. See (Dipert 1984) for a brief discussion of this history.

The argument is a good one, but it cannot be expressed as an entailment in either SL or QL1. QL can handle such arguments because the 'is the tail of' relation can be represented with a 2-place predicate.

QL has all the basic symbols of QL1, plus predicate letters for n-placed predicates, for every integer n such that $n \geq 2$.

Definition 4.1. The *basic symbols* of QL are:

(1) Logical Connectives, Punctuation Symbols, Sentence Letters, Individual Constants, Individual Variables: same as QL1

(2) 1-Place Predicates: A', B', ..., T', A'_1, B'_1, ..., T'_1, A'_2, B'_2, ...

(3) 2-Place Predicates: A'', B'', ..., T'', A''_1, B''_1, ..., T''_1, A''_2, B''_2, ...

(4) 3-Place Predicates: A''', B''', ...

. . . and so on for all positive integers.

The superscripted prime marks express the arity, or number of places, of the predicate. The subscripted integers allow for an endless supply of predicates. For every integer n, QL contains an infinite number of n-place predicates.

4.1.2 Formulas of QL

As with QL1 we must define QL formulas before getting to QL sentences. The second base clause is the only one that differs from those of the QL1 definition of formula.

Definition 4.2. The *formulas of QL* are given by the following recursive definition:

Base Clauses:

(1) A sentence letter is a formula.

(2) An n-place predicate followed by n tokens of individual constants or variables is a formula.

Generating Clauses:

(1) If ϕ is a formula then so is $\sim\phi$.

(2) If ϕ and θ are formulas then so are $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.

(3) If all of $\phi_1, \phi_2, \phi_3, \phi_4, \ldots, \phi_n$ are formulas (the list must include at least two formulas and be finite) then so are $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \ldots \vee \phi_n)$.

(4) If ϕ is a formula and it does not contain an expression of the form $\forall\alpha$ or $\exists\alpha$ for some QL variable α, then $\forall\alpha\phi$ and $\exists\alpha\phi$ are formulas.

Closure Clause: A string of symbols is a formula iff it can be generated by the clauses above.

Formulas that can be constructed from the base clauses are *atomic*. $A'\mathrm{b}$ is an atomic formula, as is $A''x\mathrm{a}$. $B''x$ is not a formula because it has a 2-place predicate followed only by one individual variable. To determine whether some string is a formula, we must count *tokens* of variables and constants. For example, $C'''xxx$ is a formula because it has a 3-place predicate followed by three variable tokens.

$G''xy$ is a formula, so by clause 4 it follows that the following are also formulas. This list is not exhaustive.

6. $\forall xG''xy$
7. $\exists xG''xy$
8. $\exists zG''xy$
9. $\exists y\forall xG''xy$
10. $\forall x\exists yG''xy$
11. $\forall x\forall zG''xy$

$\forall x\forall xG''xy$ is *not* a formula, because it's of the form $\forall x\phi$ where ϕ is a formula that contains the expression $\forall x$.

As in SL and QL1 there are unofficial formulas.

Definition 4.3. A string of symbols is an *unofficial* formula iff we can obtain it from an official formula by

(1) deleting outer parentheses,

(2) replacing official parentheses ( ) with square brackets [ ] or curly brackets { }, or

(3) omitting primes $'$ on a predicate letter.

A unique official formula can always be reconstructed from an unofficial formula.

4.1.3 Other Properties of Formulas

The concepts of subformula, order, main connective, and construction tree for formulas of QL are the same as in QL1.[2]

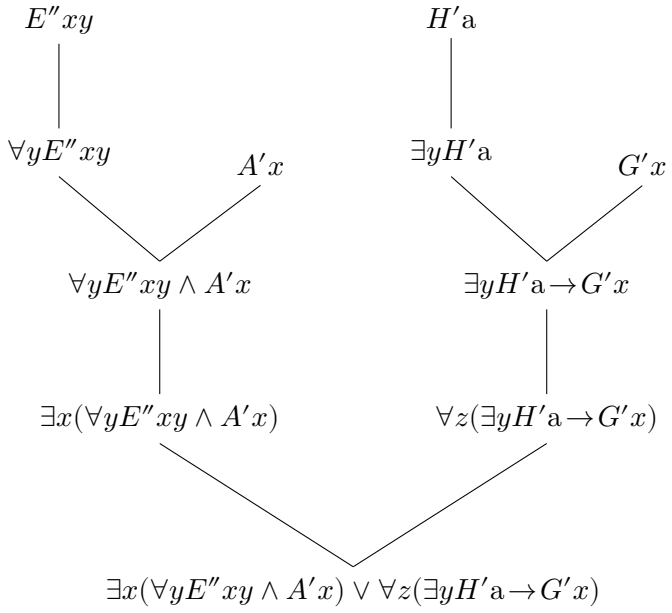
Example 4.4. Consider the formula $\exists x(\forall yE''xy \wedge A'x) \vee \forall z(\exists yH'\mathrm{a} \rightarrow G'x)$. This is a disjunction; its main connective is vee, $\vee$. It has eleven subformulas:

(1) $\exists x(\forall yE''xy \wedge A'x) \vee \forall z(\exists yH'\mathrm{a} \rightarrow G'x)$

(2) $\exists x(\forall yE''xy \wedge A'x)$
(3) $\forall yE''xy \wedge A'x$
(4) $\forall yE''xy$
(5) $A'x$
(6) $E''xy$
(7) $\forall z(\exists yH'\mathrm{a} \rightarrow G'x)$
(8) $\exists yH'\mathrm{a} \rightarrow G'x$
(9) $\exists yH'\mathrm{a}$
(10) $G'x$
(11) $H'\mathrm{a}$

[2] See section 3.1.3 of the last chapter.

The construction tree of the formula is:



As you can see from the construction tree, the order of the formula is 5.

4.1.4 Sentences of QL

Sentences, atomic sentences, and unofficial sentences of QL are defined exactly as in QL1.[3]

4.2 Models

4.2.1 Models in QL

QL models are essentially the same as models of QL1 except that they accommodate many-place predicates.

Definition 4.5. A *model* for ϕ, $\mathfrak{m}$, consists of:

(1) an assignment of a truth value `T` or `F` to each sentence letter in ϕ;

(2) a non-empty set U, called the *universe* or *domain*;

(3) an assignment of an element from U to each individual constant in ϕ;

(4) an assignment of a subset of U to each 1-place predicate in ϕ;

(5) an assignment of a set of ordered n-tuples to each n-place predicate in ϕ. The elements in each n-tuple are members of U.

[3] See section 3.1.4 of the last chapter.

Clause (5) is the only new part of the definition.

To illustrate QL models consider the 3-place predicate B'''. Let there be a model $\mathfrak{m}$ on which $\mathfrak{m}(U) = \{1, 2, 3, 4\}$ and B''' stands for the 'between' relation for positive integers. Then $\mathfrak{m}(B''')$ is a set of 3-tuples $\langle x, y, z\rangle$ such that $y < x < z$:

$$\mathfrak{m}(B''') = \{\langle 2, 1, 3\rangle, \langle 2, 1, 4\rangle, \langle 3, 1, 4\rangle, \langle 3, 2, 4\rangle\}$$

For a second illustration, consider a model $\mathfrak{m}_2$ whose domain contains exactly three people: Jack, Jill, and Bill. Let Jack be taller than Jill and Jill be taller than Bill. Let $\mathfrak{m}_2(A'')$ is the 'taller than' relation. The assignment to A'' is the following set of ordered pairs:

$$\mathfrak{m}_2(A'') = \{\langle \text{Jack, Jill}\rangle, \langle \text{Jill, Bill}\rangle, \langle \text{Jack, Bill}\rangle\}$$

As with SL and QL1, there are QL models for sets of sentences, models of QL, and QL minimal models:

Definition 4.6. $\mathfrak{m}$ is a *model for a set of QL sentences* Δ iff $\mathfrak{m}$ is a model for each sentence in Δ.

Definition 4.7. $\mathfrak{m}$ is a *model for QL* iff $\mathfrak{m}$ is a model for every sentence of QL.

Definition 4.8. $\mathfrak{m}$ is a *minimal model for* ϕ iff $\mathfrak{m}$ makes the minimum assignments necessary for $\mathfrak{m}$ to be a model for ϕ.

The proof of the minimal model theorem for QL is almost precisely the same as QL1 proof (3.36) and is left as an exercise for the reader.

4.2.2 Truth in a Model

The definition of truth in a model for QL is exactly the same as in QL1 except with an additional clause for many-place predicates.

Definition 4.9. The following clauses fix when a QL sentence θ is *true* (or *false*) on a model for θ, $\mathfrak{m}$:

(1) A sentence letter ϕ is true on $\mathfrak{m}$ iff $\mathfrak{m}(\phi) = \mathsf{T}$.

(2) An atomic sentence Pt with a 1-place predicate P and an individual term t is true on $\mathfrak{m}$ iff $\mathfrak{m}(t) \in \mathfrak{m}(P)$.

(3) An atomic sentence $Pt_1 \ldots t_n$ with an n-place predicate P is true on $\mathfrak{m}$ iff $\langle \mathfrak{m}(t_1), \mathfrak{m}(t_2), \ldots, \mathfrak{m}(t_n)\rangle \in \mathfrak{m}(P)$.

(4) A negation $\sim\phi$ is true on $\mathfrak{m}$ iff ϕ is false on $\mathfrak{m}$.

(5) A conjunction $(\phi_1 \wedge \ldots \wedge \phi_n)$ is true on $\mathfrak{m}$ iff all of $\phi_1, \ldots, \phi_n$ are true on $\mathfrak{m}$.

(6) A disjunction $(\phi_1 \vee \ldots \vee \phi_n)$ is true on $\mathfrak{m}$ iff at least one of $\phi_1, \ldots, \phi_n$ is true on $\mathfrak{m}$.

(7) A conditional $(\psi \rightarrow \phi)$ is true on $\mathfrak{m}$ iff the LHS ψ is false or the RHS ϕ is true on $\mathfrak{m}$.

(8) A biconditional $(\psi \leftrightarrow \phi)$ is true on $\mathfrak{m}$ iff ψ and ϕ have the same truth value on $\mathfrak{m}$.

(9) A universal quantification $\forall\alpha\phi$ is true on $\mathfrak{m}$ iff $\phi t/\alpha$ is true on *all* t-variants of $\mathfrak{m}$, where t is the first constant not in ϕ.

(10) An existential quantification $\exists\alpha\phi$ is true on $\mathfrak{m}$ iff $\phi t/\alpha$ is true on *some* t-variant of $\mathfrak{m}$, where t is the first constant not in ϕ.

(11) A sentence ϕ is false on $\mathfrak{m}$ iff ϕ is not true on $\mathfrak{m}$.

Clause (3) is the only new one. To see how QL truth works for a two-place predicate, consider a model $\mathfrak{m}$ with the following assignments:

$\mathfrak{m}(\mathrm{j}) = \text{Jack}$
$\mathfrak{m}(\mathrm{i}) = \text{Jill}$
$\mathfrak{m}(\mathrm{b}) = \text{Bill}$
$\mathfrak{m}(A'') = \{\langle \text{Jack, Jill}\rangle, \langle \text{Jill, Bill}\rangle, \langle \text{Jack, Bill}\rangle\}$

Example 4.10. Show that the sentence $A''\mathrm{jb}$ is true on $\mathfrak{m}$.

> To determine the truth value of $A''\mathrm{jb}$ we must check whether $\langle \mathfrak{m}(\mathrm{j}), \mathfrak{m}(\mathrm{b})\rangle$ is a member of the set $\mathfrak{m}(A'')$.

Proof: According to $\mathfrak{m}$, $\langle \mathfrak{m}(\mathrm{j}), \mathfrak{m}(\mathrm{b})\rangle = \langle \text{Jack, Bill}\rangle$. We find that $\langle \text{Jack, Bill}\rangle \in \mathfrak{m}(A'')$, so $A''\mathrm{jb}$ is true on $\mathfrak{m}$. ■

Example 4.11. Show that the sentence $A''\mathrm{bi}$ is false on $\mathfrak{m}$.

Proof: Since $\langle \mathfrak{m}(\mathrm{b}), \mathfrak{m}(\mathrm{i})\rangle = \langle \text{Bill, Jill}\rangle$ and $\langle \text{Bill, Jill}\rangle \notin \mathfrak{m}(A'')$, it follows that $A''\mathrm{bi}$ is false on $\mathfrak{m}$. ■

> Remember that $\langle \text{Bill, Jill}\rangle$ is not the same as $\langle \text{Jill, Bill}\rangle$. Order matters!

Let's try more complicated examples using the models provided in figure 4.1 (on the next page).

If you look over the many-place predicates in figure 4.1 you'll notice that we don't list explicit sets of n-tuples. Instead we provide a brief description of some relation which corresponds to some such set. This is our usual practice. Writing out all n-tuples explicitly would be tedious for the model *States* and impossible for the model *Pos Int*.

	Symbol	Model	
		Pos Int	**States**
Universe:		The set of positive integers	The set of US states (2024)
Sent. Let.:	A	true	false
	B	true	false
	C	false	true
	D	true	false
	E	true	false
	G	false	true
Constants:	a	1	Louisiana
	b	9	Maine
	c	72	Georgia
	d	3	Nebraska
	e	1	New Mexico
	f	2	Texas
1-place:	A'	all pos int	Midwestern
	B'	empty set	name with > 5 letters
	C'	even	Coastal
	D'	odd	on the Pacific Coast
	E'	prime	{Ohio}
	G'	multiple of 7	{Ohio, Alabama}
2-place:	A''	first $>$ second	share a border
	B''	are equal	first is north of second
	C''	first $= 2$ times second	first $>$ second (area)
	D''	sum of them equals 7	first $>$ second (population)
	E''	first $<$ second	first is west of second
	G''	are relatively prime	both coastal, or neither
3-place:	A'''	all equal	all same population
	B'''	first $<$ second $<$ third	first is north of others
	C'''	all odd or all even	first $>$ second $>$ third (area)
	D'''	first + second = third	first + second $>$ third (area)
	E'''	first $\times$ second = third	first is west of the others
	G'''	are all relatively prime	at least two coastal

Figure 4.1: Two QL models

Example 4.12. Determine the truth value of $\forall y \exists x A'' xy$ on model *Pos Int* (figure 4.1).

> Intuitively, this sentence can be read as 'For each y, there is some x such that $x > y$.' That is, for each positive integer there is another that is greater. Thus, we may expect that $\forall y \exists x A'' xy$ is true on *Pos Int*. This insight cannot serve as a proof that the sentence is true, but it can guide our efforts as we construct a proof according to the proper definitions.

Proof: $\forall y \exists x A'' xy$ is true on *Pos Int* iff $\exists x A'' x\mathrm{a}$ is true on all a-variants of *Pos Int* (definition of truth, $\forall$). Let there be an a-variant of *Pos Int*, *Pos Int*$^{\mathrm{a}}$, such that *Pos Int*$^{\mathrm{a}}(a) = n$, where n is an arbitrary positive integer. The sentence $\exists x A'' x\mathrm{a}$ is true on *Pos Int*$^{\mathrm{a}}$ iff $A''\mathrm{ba}$ is true on some b-variant of *Pos Int*$^{\mathrm{a}}$ (definition of truth, $\exists$). *Pos Int*(A'') is the set of ordered pairs $\langle x, y\rangle$ such that $x > y$. Let *Pos Int*$^{\mathrm{ab}}$ be a b-variant of *Pos Int*$^{\mathrm{a}}$ such that *Pos Int*$^{\mathrm{ab}}(b) = n+1$. Clearly $\langle n+1, n\rangle \in$ *Pos Int*(A''). Hence, regardless of what n is, $\exists x A'' x\mathrm{a}$ is true on *Pos Int*$^{\mathrm{a}}$. Because nothing about this argument depends upon a specific assignment to a, it holds for all a-variants of *Pos Int*. Therefore $\forall y \exists x A'' xy$ is true on *Pos Int*. ■

Example 4.13. Determine the truth value of $\exists x \forall y A'' xy$ on model *Pos Int* (figure 4.1).

> Intuitively, this sentence can be read as 'There is some x such that for all y, $x > y$.' That is, there is some positive integer that is greater than all others. This is false, so we expect that $\exists x \forall y A'' xy$ is false on *Pos Int*. But, as before, intuition is no substitute for proof.

Proof: $\exists x \forall y A'' xy$ is true on *Pos Int* iff $\forall y A''\mathrm{a}y$ is true on some a-variant of *Pos Int*. Let there be an a-variant of *Pos Int*, *Pos Int*$^{\mathrm{a}}$, such that *Pos Int*$^{\mathrm{a}}(a) = n$, where n is an arbitrary positive integer. The sentence $\forall y A''\mathrm{a}y$ is true on *Pos Int*$^{\mathrm{a}}$ iff $A''\mathrm{ab}$ is true on all b-variants of *Pos Int*$^{\mathrm{a}}$. *Pos Int*(A'') is the set of ordered pairs $\langle x, y\rangle$ such that $x > y$. Let *Pos Int*$^{\mathrm{ab}}$ be a b-variant of *Pos Int*$^{\mathrm{a}}$ such that *Pos Int*$^{\mathrm{ab}}(b) = n+1$. Clearly $\langle n, n+1\rangle \notin$ *Pos Int*(A''). Hence, regardless of what n is, $A''\mathrm{ab}$ is false on *Pos Int*$^{\mathrm{ab}}$. Thus, $\forall y A''\mathrm{a}y$ is false on all a-variants of *Pos Int*. Therefore, $\exists x \forall y A'' xy$ is false on *Pos Int*. ■

The only difference between $\forall y \exists x A'' xy$ and $\exists x \forall y A'' xy$ is the order of their quantifiers. That change is enough to transform the meaning of the sentence completely. Quantifier order matters!

Example 4.14. Show that $\forall x \forall y \forall z((Cxy \wedge Dxyz) \rightarrow Byxz)$ is true on the model *Pos Int*.

Proof: $\forall x \forall y \forall z((Cxy \wedge Dxyz) \rightarrow Byxz)$ is true on *Pos Int* iff
$\forall y \forall z((C\mathrm{a}y \wedge D\mathrm{a}yz) \rightarrow B\mathrm{y}\mathrm{a}z)$ is true on all a-variants of *Pos Int*, *Pos Int*$^{\mathrm{a}}$.

$\forall y\forall z((C\mathrm{a}y \wedge D\mathrm{a}yz) \rightarrow B\mathrm{a}yz)$ is true on some *Pos Int*$^{\mathrm{a}}$ iff $\forall z((C\mathrm{ab} \wedge D\mathrm{ab}z) \rightarrow B\mathrm{ba}z)$ is true on all b-variants of *Pos Int*$^{\mathrm{a}}$, *Pos Int*$^{\mathrm{ab}}$. $\forall z((C\mathrm{ab} \wedge D\mathrm{ab}z) \rightarrow B\mathrm{ba}z)$ is true on some *Pos Int*$^{\mathrm{ab}}$ iff $((C\mathrm{ab} \wedge D\mathrm{abc}) \rightarrow B\mathrm{bac})$ is true on all c-variants of *Pos Int*$^{\mathrm{ab}}$, *Pos Int*$^{\mathrm{abc}}$. So, if there is some *Pos Int*$^{\mathrm{abc}}$ such that $((C\mathrm{ab} \wedge D\mathrm{abc}) \rightarrow B\mathrm{bac})$ is false, then $\forall x\forall y\forall z((Cxy \wedge Dxyz) \rightarrow Byxz)$ is false on *Pos Int*; otherwise, it's true.
Assume for indirect proof that there are assignments to a, b, and c such that Cab and Dabc are true and Bbac is false. Let *Pos Int*$^{\mathrm{abc}}(b) = n$ for some arbitrary positive integer n. Since *Pos Int*(C) is the set of positive integer pairs $\langle u, v\rangle$ such that $u = 2v$, and Cab is true on *Pos Int*$^{\mathrm{abc}}$, it follows that *Pos Int*$^{\mathrm{abc}}(a) = 2n$. And since *Pos Int*(D) is the set of positive integer triples $\langle u, v, w\rangle$ such that $u + v = w$, and Dabc is true on *Pos Int*$^{\mathrm{abc}}$, it follows that *Pos Int*$^{\mathrm{abc}}(c) = n + 2n = 3n$. Since *Pos Int*$(B)$ is the set of positive integer triples $\langle v, u, w\rangle$ such that $v < u < w$, then $\langle \mathrm{b}, \mathrm{a}, \mathrm{c}\rangle = \langle n, 2n, 3n\rangle$. And clearly $\langle n, 2n, 3n\rangle \in$ *Pos Int*(B). It follows that Bbac is true on *Pos Int*$^{\mathrm{abc}}$. But we had assumed that Bbac is false on *Pos Int*$^{\mathrm{abc}}$. $\perp$
So there is no *Pos Int*$^{\mathrm{abc}}$ such that Cab and Dabc are true and Bbac is false. Therefore $\forall x\forall y\forall z((Cxy \wedge Dxyz) \rightarrow Byxz)$ is true on *Pos Int*. ■

Example 4.15. Show that $\forall x\forall y\forall z((Cxy \wedge Dxyz) \rightarrow Byxz)$ is false on the model *States*.

Proof: As in the last example, assume for indirect proof that there are assignments to a, b, and c such that Cab and Dabc are true and Bbac is false. Let *States*$^{\mathrm{abc}}(a) =$ Alaska, *States*$^{\mathrm{abc}}(b) =$ Delaware, and *States*$^{\mathrm{abc}}(c) =$ Rhode Island. Alaska has an area of approximately 1.7×10^6 km^2, Delaware an area of approximately 2.5×10^3 km^2, and Rhode Island an area of approximately 1.5×10^3 km^2. Hence *States*$^{\mathrm{abc}}$(a) $>$ *States*$^{\mathrm{abc}}$(b) (area), *States*$^{\mathrm{abc}}$(a) + *States*$^{\mathrm{abc}}$(b) $>$ *States*$^{\mathrm{abc}}$(c) (area), but *States*$^{\mathrm{abc}}$(b) is not north of both *States*$^{\mathrm{abc}}$(a) and *States*$^{\mathrm{abc}}$(c); that is, Delaware is not north of Alaska. So, by clause (3) of the definition of truth, 4.9, Cab and Dabc are true on *States*$^{\mathrm{abc}}$, while Bbac is false on *States*$^{\mathrm{abc}}$. So by the definition of truth ($\wedge$ and $\rightarrow$), $((C\mathrm{ab} \wedge D\mathrm{abc}) \rightarrow B\mathrm{bac})$ is false on *States*$^{\mathrm{abc}}$.

There is a c-variant of *States*$^{\mathrm{ab}}$ on which $((C\mathrm{ab} \wedge D\mathrm{abc}) \rightarrow B\mathrm{bac})$ is false. It follows (by the definition of truth for $\forall$) that $\forall z((C\mathrm{ab} \wedge D\mathrm{ab}z) \rightarrow B\mathrm{ba}z)$ is false on *States*$^{\mathrm{ab}}$. So, there is, in turn, a b-variant of *States*$^{\mathrm{a}}$ on which $\forall z((C\mathrm{ab} \wedge D\mathrm{ab}z) \rightarrow B\mathrm{ba}z)$ is false. So, again, by the definition of truth for $\forall$, that $\forall y\forall z((C\mathrm{a}y \wedge D\mathrm{a}yz) \rightarrow B\mathrm{a}yz)$ is false on *States*$^{\mathrm{a}}$.

Finally, because there is an a-variant of *States* on which $\forall y\forall z((C\mathrm{a}y \wedge D\mathrm{a}yz) \rightarrow B\mathrm{a}yz)$ is false, $\forall x\forall y\forall z((Cxy \wedge Dxyz) \rightarrow Byxz)$ is false on *States* (definition of truth, $\forall$). ■

4.2.3 Logical Truth: QT, QF, & QC

The concepts of quantificational truth (QT), quantificational falsehood (QF), and quantificational contingency (QC) are defined for QL exactly as in QL1.[4]

[4] See section 3.2.4 on page 75.

4.2.4 Entailment and other Relations

The concepts for entailment and the other logical relations are also defined for QL exactly as in QL1.[5]

4.3 The Dragnet Theorem

In example 3.46 we established that $\forall xGx \models G\mathrm{b}$ holds by reasoning as follows:

(i) Any model $\mathfrak{m}$ that makes $\forall xGx$ true makes $G\mathrm{a}$ true on all a-variants of $\mathfrak{m}$.

(ii) All a-variants of $\mathfrak{m}$ share the same domain as $\mathfrak{m}$, so there is some a-variant of $\mathfrak{m}$, $\mathfrak{m}^{\mathrm{a}}$, such that $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) = \mathfrak{m}(\mathrm{b})$.

(iii) Because $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}^{\mathrm{a}}(G)$ and $\mathfrak{m}^{\mathrm{a}}(G) = \mathfrak{m}(G)$, $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}(G)$.

(iv) Because $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) \in \mathfrak{m}(G)$ and $\mathfrak{m}^{\mathrm{a}}(\mathrm{a}) = \mathfrak{m}(\mathrm{b})$, $\mathfrak{m}(\mathrm{b}) \in \mathfrak{m}(G)$. Thus, $G\mathrm{b}$ is true on $\mathfrak{m}$.

By analogous reasoning we can show that $\forall xGx$ entails $G\mathrm{c}$, $G\mathrm{d}$, $G\mathrm{e}$, and so on. But it seems as if we should be able to save ourselves some work by proving a more general result. Can we show that the entailment holds regardless of which constant follows G? Yes, but demonstrating this is easier if we first develop QL metatheory a bit more. For such proofs we use a metavariable, in this case t: $\forall xGx \models Gx(t/x)$, where t is any constant. Remember from section 3.26, on page 69 that $\phi t/x$ is the result of replacing every unbound token of x with a token of t in ϕ.

We also want to prove stronger entailments, such as: $\forall x\phi \models \phi\mathrm{b}/x$, where ϕ is some QL formula with only x free.

To simplify the task of proving such general theorems we use the 'Dragnet Theorem'. Informally, the theorem says: "If you have a true sentence and replace the constants in it, but keep the same elements assigned to the new constants, you get another true sentence." Dragnet is extraordinarily helpful for proving many significant results about QL (e.g., in theorem 8.9, on page 183, and 8.25, on page 199). Although the claim behind Dragnet is fairly straightforward, its proof is long and complicated.

> The name "Dragnet" comes from the Dragnet radio and TV show. The theorem bears a close resemblance to the show's opening narration: "The story you are about to see is true. Only the names have been changed to protect the innocent." – Sergeant Joe Friday, LAPD.

Theorem 4.16. The Dragnet Theorem: If

[5] See section 3.3 on page 77.

(1) there exists:
(a) a list of distinct constants t_1, t_2, ..., t_i, s_1, s_2, ..., s_i,
(b) a QL sentence ϕ that contains t_1, t_2, ..., t_i but not s_1, s_2, ..., s_i,
(c) a second sentence $\phi^* = \phi s_1/t_1, s_2/t_2, \ldots, s_i/t_i$, and

(2) (a) $\mathfrak{m}_1$ and $\mathfrak{m}_2$ are QL models that have the same domain U and make the same assignments to everything in ϕ except t_1, t_2, ..., t_i, and
(b) $\mathfrak{m}_1(t_1) = \mathfrak{m}_2(s_1)$, $\mathfrak{m}_1(t_2) = \mathfrak{m}_2(s_2)$, ..., $\mathfrak{m}_1(t_i) = \mathfrak{m}_2(s_i)$,

Then: ϕ is true on $\mathfrak{m}_1$ iff ϕ^* is true on $\mathfrak{m}_2$.[6]

Proof:

In this proof we treat * as a function that takes a QL formula ϕ with constants t_1, t_2, ..., t_i and returns an otherwise identical formula in which those constants are replaced with ones not in ϕ, s_1, s_2, ..., s_i, respectively: $\phi^* = \phi s_1/t_1, s_2/t_2, \ldots, s_i/t_i$. This stipulation allows us to take for granted that any pair of sentences ψ and ψ^* satisfies Dragnet condition (1) above. Additionally, throughout the proof we assume that $\mathfrak{m}_1$ and $\mathfrak{m}_2$ are two arbitrary models that satisfy Dragnet condition (2) above.

Base Step: ϕ is atomic. There are three cases:

(1) ϕ is a sentence letter. Then there are no constants and ϕ and ϕ^* are be identical. Clearly ϕ is true on $\mathfrak{m}_1$ iff ϕ^* is true on $\mathfrak{m}_2$.

(2) ϕ is a predicate letter P followed by n constants, $Pq_1q_2 \ldots q_n$. Some or all of these constants are to be replaced in ϕ^*. We label the constants to be replaced $t_1, t_2, \ldots, t_i$, and their replacements $s_1, s_2, \ldots, s_i$. Then $\phi^* = \phi s_1/t_1, s_2/t_2, \ldots, s_i/t_i$. Let's assume, without loss of generality, that $\phi = Pq_1 \ldots t_1 \ldots t_2 \ldots t_i \ldots q_n$. So ϕ^* is $Pq_1 \ldots s_1 \ldots s_2 \ldots s_i \ldots q_n$. By the definition of truth,

$$\phi \text{ is true on } \mathfrak{m}_1 \text{ iff } \langle \mathfrak{m}_1(q_1), \ldots, \mathfrak{m}_1(t_1), \ldots, \mathfrak{m}_1(t_i), \ldots, \mathfrak{m}_1(q_n)\rangle \in \mathfrak{m}_1(P).$$

By Dragnet condition (2), $\mathfrak{m}_1(P) = \mathfrak{m}_2(P)$, so we substitute:

$$\phi \text{ is true on } \mathfrak{m}_1 \text{ iff } \langle \mathfrak{m}_1(q_1), \ldots, \mathfrak{m}_1(t_1), \ldots, \mathfrak{m}_1(t_i), \ldots, \mathfrak{m}_1(q_n)\rangle \in \mathfrak{m}_2(P).$$

By Dragnet condition (2), $\mathfrak{m}_1(t_1) = \mathfrak{m}_2(s_1)$, ..., $\mathfrak{m}_1(t_i) = \mathfrak{m}_2(s_i)$, so we again substitute:

$$\phi \text{ is true on } \mathfrak{m}_1 \text{ iff } \langle \mathfrak{m}_1(q_1), \ldots, \mathfrak{m}_2(s_1), \ldots, \mathfrak{m}_2(s_i), \ldots, \mathfrak{m}_1(q_n)\rangle \in \mathfrak{m}_2(P).$$

$\mathfrak{m}_1$ and $\mathfrak{m}_2$ otherwise make all the same assignments. So for each constant q_k in ϕ that *isn't* replaced in ϕ^*, $\mathfrak{m}_1(q_k) = \mathfrak{m}_2(q_k)$:

[6] See Mates 1972: 66 and Bergmann et al. 2003: 577 for different versions of essentially the same theorem.

ϕ is true on $\mathfrak{m}_1$ iff $\langle \mathfrak{m}_2(q_1), \ldots, \mathfrak{m}_2(s_1), \ldots, \mathfrak{m}_2(s_i), \ldots, \mathfrak{m}_2(q_n)\rangle \in \mathfrak{m}_2(P)$.

Thus, by the definition of truth:

ϕ is true on $\mathfrak{m}_1$ iff ϕ^* is true on $\mathfrak{m}_2$.

Inheritance Step:

Recursive Assumption Assume that the Dragnet property holds of all QL sentences of order k or less and that ϕ is an QL sentence of order $k+1$. Consider all the ways to construct a sentence of order $k+1$:

Negation: ϕ is of the form $\sim\psi$. Since the symbol '$\sim$' is not a constant it is not replaced in ϕ^*. Then $\phi^* = \sim\psi^*$. Since ψ is of order k, by recursive assumption (RA):

ψ is true on $\mathfrak{m}_1$ iff ψ^* is true on $\mathfrak{m}_2$,

It follows that:

ψ is false on $\mathfrak{m}_1$ iff ψ^* is false on $\mathfrak{m}_2$.

ψ is false on $\mathfrak{m}_1$ iff $\sim\psi$ is true on $\mathfrak{m}_1$. The same holds of ψ^*. Thus:

$\sim\psi$ is true on $\mathfrak{m}_1$ iff $\sim\psi^*$ is true on $\mathfrak{m}_2$.

Conjunction: ϕ is of the form $(\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n)$. Since the symbol '$\wedge$' is not a constant these tokens are not replaced in ϕ^*. Then $\phi^* = (\psi_1^* \wedge \psi_2^* \wedge \ldots \wedge \psi_n^*)$. Let ψ_j be the j^{th} conjunct of ϕ, where $1 \leq j \leq n$. Each ψ_j is of order k or lower. So, by RA:

For all ψ_j, ψ_j is true on $\mathfrak{m}_1$ iff ψ_j^* is true on $\mathfrak{m}_2$.

From these 'iff' clauses it follows that:

All of ψ_1, ψ_2, ..., ψ_n are true on $\mathfrak{m}_1$ iff all of ψ_1^*, ψ_2^*, ..., ψ_n^* are true on $\mathfrak{m}_2$.

Thus, by the definition of truth for $\wedge$:

$(\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n)$ is true on $\mathfrak{m}_1$ iff $(\psi_1^* \wedge \psi_2^* \wedge \ldots \wedge \psi_n^*)$ is true on $\mathfrak{m}_2$.

Disjunction: We leave this case for the reader as an exercise.

Conditional: ϕ is of the form $(\psi \rightarrow \theta)$. Since the symbol '$\rightarrow$' is not a constant it is not replaced in ϕ^*. Then $(\psi \rightarrow \theta)^* = (\psi^* \rightarrow \theta^*)$. By the definition of truth for $\rightarrow$:

$(\psi \rightarrow \theta)$ is true on $\mathfrak{m}_1$ iff either (i) ψ is false on $\mathfrak{m}_1$, or (ii) θ is true on $\mathfrak{m}_1$.

The sentence ψ is of order k or lower. Then, by RA, ψ is true on $\mathfrak{m}_1$ iff ψ^* is true on $\mathfrak{m}_2$. It follows that ψ is false on $\mathfrak{m}_1$ iff ψ^* is false on $\mathfrak{m}_2$. So we substitute into part (i):

$(\psi \to \theta)$ is true on $\mathfrak{m}_1$ iff either (i) ψ^* is false on $\mathfrak{m}_2$, or (ii) θ is true on $\mathfrak{m}_1$.

Similarly, θ is of order k or lower. Then by RA, θ is true on $\mathfrak{m}_1$ iff θ^* is true on $\mathfrak{m}_2$. So we substitute into part (ii):

$(\psi \to \theta)$ is true on $\mathfrak{m}_1$ iff either (i) ψ^* is false on $\mathfrak{m}_2$, or (ii) θ^* is true on $\mathfrak{m}_2$.

By the definition of truth for $\to$:

$$(\psi \to \theta) \text{ is true on } \mathfrak{m}_1 \text{ iff } (\psi^* \to \theta^*) \text{ is true on } \mathfrak{m}_2.$$

And since $(\psi \to \theta)^* = (\psi^* \to \theta^*)$:

$$(\psi \to \theta) \text{ is true on } \mathfrak{m}_1 \text{ iff } (\psi \to \theta)^* \text{ is true on } \mathfrak{m}_2.$$

Biconditional: We leave this case for the reader as an exercise.

Quantifier Preface: ϕ is of the form $\forall\beta\psi$ or $\exists\beta\psi$. Then formula ψ is of order k. To reduce the work needed for the quantifier clauses, we first prove an intermediate result. Let q be the first constant not in ψ and let r be the first constant not in ψ^*.

> We cannot assume that $q = r$. Consider the case such that $\psi = (Dx \to Gx\mathrm{b})$ and $\psi^* = \psi\mathrm{a/b} = (Dx \to Gx\mathrm{a})$. The first constant not in $(Dx \to Gx\mathrm{b})$ is a, and the first constant not in $(Dx \to Gx\mathrm{a})$ is b.

Since $\mathfrak{m}_1$ and $\mathfrak{m}_2$ satisfy Dragnet condition (2), $\mathfrak{m}_1(U) = \mathfrak{m}_2(U)$. Let Δ be the domain shared by $\mathfrak{m}_1$ and $\mathfrak{m}_2$. Then:

For each $\delta \in \Delta$ there is a q-variant of $\mathfrak{m}_1$, $\mathfrak{m}_1^{\mathrm{q}}(q) = \delta$, and an r-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{r}}(r) = \delta$, such that $\psi q/\beta$ is true on $\mathfrak{m}_1^{\mathrm{q}}$ iff $\psi^* r/\beta$ is true on $\mathfrak{m}_2^{\mathrm{r}}$.

> The purpose of this claim may not be obvious. If you are willing to accept on faith that it's a useful result, feel free to continue. If not, then skip down to the 'Universal Quantification' clause and read through that until you come to the step that cites the Quantifier Preface. Once you understand why this Quantifier Preface is helpful, then come back and pick up where you left off.

Subproof:

It may be tempting to try to derive this result by applying the recursive assumption directly to $\psi q/\beta$ and $\psi^* r/\beta$. However, it's possible that q is in $\psi r/\beta$ or r is in $\psi q/\beta$. Either one violates Dragnet condition (1).

Our strategy is to use a third sentence, $\psi^* u/\beta$, where u is a variable that is not in either $\psi q/\beta$ or $\psi^* r/\beta$. We first prove the desired result with $\psi^* r/\beta$ and $\psi^* u/\beta$ (i.e. that there are variants of $\mathfrak{m}_1/\mathfrak{m}_2$ such that one sentence is true iff the other is). Then we do the same with $\psi q/\beta$ and $\psi^* u/\beta$. Finally, we can 'factor out' $\psi^* u/\beta$ as a middle term, and prove the desired result for $\psi q/\beta$ and $\psi^* r/\beta$.

(Part 1:) Let u be a constant not in either $\psi^* r/\beta$ or $\psi^* u/\beta$. Then $\psi^* r/\beta$ and $\psi^* u/\beta$ are the same except that the latter sentence has u in place of r. These sentences satisfy Dragnet condition (1).

Let δ be an arbitrary element of Δ. Since $\Delta = \mathfrak{m}_2(U)$, $\delta \in \mathfrak{m}_2(U)$, and so there is an r-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{r}}$, such that $\mathfrak{m}_2^{\mathrm{r}}(r) = \delta$. And there is a u-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{u}}$, such that $\mathfrak{m}_2^{\mathrm{u}}(u) = \delta$. So, $\mathfrak{m}_2^{\mathrm{r}}$ and $\mathfrak{m}_2^{\mathrm{u}}$ make the same assignments to everything in $\psi^* r/\beta$ besides r, and $\mathfrak{m}_2^{\mathrm{u}} = \mathfrak{m}_2^{\mathrm{r}}$. These models satisfy Dragnet condition (2).

$\psi^* r/\beta$ and $\psi^* u/\beta$ are of order k. Thus, by RA:

For each $\delta \in \Delta$ there is a r-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{r}}(q) = \delta$, and a u-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{u}}(u) = \delta$, such that $\psi^* r/\beta$ is true on $\mathfrak{m}_2^{\mathrm{r}}$ iff $\psi^* u/\beta$ is true on $\mathfrak{m}_2^{\mathrm{u}}$.

(Part 2:) The formula ψ contains the terms $t_1, t_2, \ldots, t_i, \beta$. Then $\psi q/\beta$ is a sentence that contains the constants $t_1, t_2, \ldots, t_i, q$. And $\psi^* u/\beta$ is a sentence that contains the constants $s_1, s_2, \ldots, s_i, u$. It follows that $\psi q/\beta$ and $\psi^* u/\beta$ are exactly alike, except that $t_1, t_2, \ldots, t_i, q$ in $\psi q/\beta$ are replaced with $s_1, s_2, \ldots, s_i, u$ in $\psi^* u/\beta$. So the sentences $\psi q/\beta$ and $\psi^* u/\beta$ satisfy Dragnet condition (1).

Let δ be an arbitrary element of Δ. Since $\Delta = \mathfrak{m}_1(U)$, $\delta \in \mathfrak{m}_1(U)$, and so there is a q-variant of $\mathfrak{m}_1$, $\mathfrak{m}_1^{\mathrm{q}}$, such that $\mathfrak{m}_1^{\mathrm{q}}(q) = \delta$. And since $\Delta = \mathfrak{m}_2(U)$, $\delta \in \mathfrak{m}_2(U)$, and so there is a u-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{u}}$, such that $\mathfrak{m}_2^{\mathrm{u}}(u) = \delta$. Hence $\mathfrak{m}_1^{\mathrm{q}}(q) = \mathfrak{m}_2^{\mathrm{u}}(u)$.

It was assumed that $\mathfrak{m}_1(t_1) = \mathfrak{m}_2(s_1)$, $\mathfrak{m}_1(t_2) = \mathfrak{m}_2(s_2)$, ..., $\mathfrak{m}_1(t_i) = \mathfrak{m}_2(s_i)$. The model $\mathfrak{m}_1$ differs from $\mathfrak{m}_1^{\mathrm{q}}$ only on the assignment to q. But q is the first constant not in ψ, so $\mathfrak{m}_1$ and $\mathfrak{m}_1^{\mathrm{q}}$ make the same assignments to each of t_1, ..., t_i. Analogous reasoning shows that $\mathfrak{m}_2$ and $\mathfrak{m}_2^{\mathrm{u}}$ make the same assignments to each of s_1, ..., s_i. Therefore, $\mathfrak{m}_1^{\mathrm{q}}(t_1) = \mathfrak{m}_2^{\mathrm{u}}(s_1)$, $\mathfrak{m}_1^{\mathrm{q}}(t_2) = \mathfrak{m}_2^{\mathrm{u}}(s_2)$, ..., $\mathfrak{m}_1^{\mathrm{q}}(t_i) = \mathfrak{m}_2^{\mathrm{u}}(s_i)$, and $\mathfrak{m}_1^{\mathrm{q}}(q) = \mathfrak{m}_2^{\mathrm{u}}(u)$.

And since $\mathfrak{m}_1^{\mathrm{q}}$ and $\mathfrak{m}_2^{\mathrm{u}}$ are variants of $\mathfrak{m}_1$ and $\mathfrak{m}_2$, respectively, they make the

same assignments to everything in $\psi q/\beta$ apart from $t_1, t_2, \ldots, t_i, q$. Thus, $\mathfrak{m}_1^{\mathrm{q}}$ and $\mathfrak{m}_2^{\mathrm{u}}$ satisfy Dragnet condition (2). The sentences $\psi q/\beta$ and $\psi^* u/\beta$ are each of order k, so, by RA,

For each $\delta \in \Delta$ there is a q-variant of $\mathfrak{m}_1$, $\mathfrak{m}_1^{\mathrm{q}} = \delta$, and a u-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{u}} = \delta$, such that $\psi q/\beta$ is true on $\mathfrak{m}_1^{\mathrm{q}}$ iff $\psi^* u/\beta$ is true on $\mathfrak{m}_2^{\mathrm{u}}$.

From the conclusions of parts 1 and 2, it therefore follows that:

For each $\delta \in \Delta$ there is a q-variant of $\mathfrak{m}_1$, $\mathfrak{m}_1^{\mathrm{q}} = \delta$, and an r-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{r}} = \delta$, such that $\psi q/\beta$ is true on $\mathfrak{m}_1^{\mathrm{q}}$ iff $\psi^* r/\beta$ is true on $\mathfrak{m}_2^{\mathrm{r}}$.

■

Universal Quantification: ϕ is of the form $\forall\beta\psi$, where ψ is a formula that has exactly one free variable, β. Since neither '$\forall$' nor 'β' is a constant, $\phi^* = \forall\beta\psi^*$. Since $\mathfrak{m}_1$ and $\mathfrak{m}_2$ satisfy Dragnet condition (2), $\mathfrak{m}_1(U) = \mathfrak{m}_2(U)$. Call this shared domain Δ. By the Quantifier Preface, for each $\delta \in \Delta$ there is a q-variant of $\mathfrak{m}_1$, $\mathfrak{m}_1^{\mathrm{q}}$, and an r-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{r}}$, such that $\psi q/\beta$ is true on $\mathfrak{m}_1^{\mathrm{q}}$ iff $\psi^* r/\beta$ is true on $\mathfrak{m}_2^{\mathrm{r}}$. From this it follows that:

$\psi q/\beta$ is true on every q-variant of $\mathfrak{m}_1$ iff
$\psi^* r/\beta$ is true on every r-variant of $\mathfrak{m}_2$.

By the definition of truth of $\forall$:

$\forall\beta\psi$ is true on $\mathfrak{m}_1$ iff $\psi q/\beta$ is true on every q-variant of $\mathfrak{m}_1$.

And:

$\forall\beta\psi^*$ is true on $\mathfrak{m}_2$ iff $\psi^* r/\beta$ is true on every r-variant of $\mathfrak{m}_2$.

Substitute these two results into the first to get:

$\forall\beta\psi$ is true on $\mathfrak{m}_1$ iff $\forall\beta\psi^*$ is true on $\mathfrak{m}_2$.

Existential Quantification: ϕ is of the form $\exists\beta\psi$, where ψ is a formula that has exactly one free variable, β. Since neither '$\exists$' nor 'β' is a constant, $\phi^* = \exists\beta\psi^*$. Since $\mathfrak{m}_1$ and $\mathfrak{m}_2$ satisfy Dragnet condition (2), $\mathfrak{m}_1(U) = \mathfrak{m}_2(U)$. Call this shared domain Δ. By the Quantifier Preface, for each $\delta \in \Delta$ there is a q-variant of $\mathfrak{m}_1$, $\mathfrak{m}_1^{\mathrm{q}}$, and an r-variant of $\mathfrak{m}_2$, $\mathfrak{m}_2^{\mathrm{r}}$, such that $\psi q/\beta$ is true on $\mathfrak{m}_1^{\mathrm{q}}$ iff $\psi^* r/\beta$ is true on $\mathfrak{m}_2^{\mathrm{r}}$. From this it follows that:

$\psi q/\beta$ is true on some q-variant of $\mathfrak{m}_1$ iff
$\psi^* r/\beta$ is true on some r-variant of $\mathfrak{m}_2$.

By the definition of truth of $\exists$:

$\exists\beta\psi$ is true on $\mathfrak{m}_1$ iff $\psi q/\beta$ is true on some q-variant of $\mathfrak{m}_1$.

And:

$\exists\beta\psi^*$ is true on $\mathfrak{m}_2$ iff $\psi^* r/\beta$ is true on some r-variant of $\mathfrak{m}_2$.

Substitute these two results into the first to get:

$\exists\beta\psi$ is true on $\mathfrak{m}_1$ iff $\exists\beta\psi^*$ is true on $\mathfrak{m}_2$.

Closure Step: There is no other way to construct a QL sentence, so we have shown that the Dragnet property holds of all QL sentences.

■

Dragnet is particularly useful when proving theorems about sentence schemas, i.e., sentences of a general form but with inessential details left unspecified. Most logic texts do not prove Dragnet but prove results using specific cases of Dragnet as needed. In effect, we have done that work once and for all by proving Dragnet in fully general form. We prefer to state the full proof to make its pattern evident.

Example 4.17. Let ϕ be a formula whose only free variable is x. Prove that $\forall x\phi \models \phi\mathrm{b}/x$.

Proof: Suppose $\forall x\phi$ is true on $\mathfrak{m}$. Then every t-variant of $\mathfrak{m}$ makes $\phi t/x$ true. There is a t-variant, $\mathfrak{m}^{\mathrm{t}}$, that makes the same assignment to t that $\mathfrak{m}$ assigns to b. So $\phi t/x$ is true on $\mathfrak{m}^{\mathrm{t}}$. The sentences $\phi t/x$ and $\phi\mathrm{b}/x$ are the same, except that t is replaced by b in the latter. So they satisfy Dragnet condition (1). The models $\mathfrak{m}$ and $\mathfrak{m}^{\mathrm{t}}$ make the same assignments to everything in $\phi t/x$ besides t, and $\mathfrak{m}^{\mathrm{t}}(t) = \mathfrak{m}(\mathrm{b})$. So they satisfy Dragnet condition (2). Thus, by Dragnet, $\mathfrak{m}$ makes $\phi\mathrm{b}/x$ true. Therefore $\forall x\phi \models \phi\mathrm{b}/x$. ■

When might the Dragnet theorem be useful? Pay attention to the two Dragnet conditions. When you can show that those hold there is a decent chance Dragnet can help. In some cases you will need to be strategic in your selection of a model variant.

While the Dragnet theorem is helpful and illustrates an important general pattern it can be a bit unwieldy. So we prove another, the 'Free Choice' theorem, which builds on Dragnet and saves us some hassle.

Theorem 4.18. The Free Choice Theorem: (i) A QL sentence of the form $\forall\alpha\phi$ is true on some model $\mathfrak{m}$ iff all s-variants of $\mathfrak{m}$ make $\phi s/\alpha$ true, where s is any constant not in ϕ; and (ii) a QL sentence of the form $\exists\alpha\phi$ is true on some model $\mathfrak{m}$ iff some s-variant of $\mathfrak{m}$ makes $\phi s/\alpha$ true, where s is any constant not in ϕ.

Proof: (i) Let ϕ be a formula with only α free. Let t be the first constant not in ϕ and let s be an arbitrary constant not in ϕ. There are two cases. (Case: 1) $s = t$. By the definition of truth for $\forall$, $\forall\alpha\phi$ is true on $\mathfrak{m}$ iff all s-variants of $\mathfrak{m}$ make $\phi s/\alpha$ true.

(Case: 2) $s \neq t$. The sentences $\phi t/\alpha$ and $\phi s/\alpha$ are the same, except $\phi s/\alpha$ has s where $\phi t/\alpha$ has t. So they satisfy condition (1) of Dragnet.

Let $\mathfrak{m}$ be some model for $\forall\alpha\phi$. All variants of $\mathfrak{m}$ share a domain. So for each element of $\mathfrak{m}(U)$, there is a t-variant of $\mathfrak{m}$ and a corresponding s-variant of $\mathfrak{m}$ that assign this element to t and s, respectively. Let $\mathfrak{m}^{\mathrm{t}}$ and $\mathfrak{m}^{\mathrm{s}}$ be variants such that $\mathfrak{m}^{\mathrm{t}}(t) = \mathfrak{m}^{\mathrm{s}}(s)$. They're both variants of $\mathfrak{m}$, so they make the same assignments to everything in $\phi t/\alpha$ besides t. Thus $\mathfrak{m}^{\mathrm{t}}$ and $\mathfrak{m}^{\mathrm{s}}$ satisfy condition (2) of Dragnet.

By Dragnet, $\phi t/\alpha$ is true on $\mathfrak{m}$ iff $\phi s/\alpha$ is true on $\mathfrak{m}^s$. We assumed nothing special about s except that it's a constant not in ϕ. So, all t-variants of $\mathfrak{m}$ make $\phi t/\alpha$ true iff all s-variants of $\mathfrak{m}$ make $\phi s/\alpha$ true. Therefore, by the definition of truth of $\forall$, $\forall\alpha\phi$ is true on $\mathfrak{m}$ iff all s-variants of $\mathfrak{m}$ make $\phi s/\alpha$ true.

(ii) We leave this part as an exercise for the reader. ■

Example 4.19. Let ϕ and ψ be formulas whose only free variable is x. Prove that $\forall x(\phi\rightarrow\psi) \models \forall x\phi\rightarrow\forall x\psi$.

Proof: Let $\mathfrak{m}$ be some model that makes $\forall x(\phi\rightarrow\psi)$ true. Then, by the definition of truth for $\forall$, $(\phi\rightarrow\psi)t/x$ is true on all t-variants of $\mathfrak{m}$. Since ϕ and ψ are parts of $\phi\rightarrow\psi$, and the first constant not in $\phi\rightarrow\psi$ is t, t is not in ϕ or ψ. There are two cases.

(Case 1:) $\phi t/x$ is false on at least one t-variant of $\mathfrak{m}$. Then, by the Free Choice theorem, $\forall x\phi$ is false on $\mathfrak{m}$, and thus, by the definition of truth for $\rightarrow$, $\forall x\phi\rightarrow\forall x\psi$ is true on $\mathfrak{m}$.

> We cannot directly use the definition of truth to conclude that $\forall x\phi$ is false on $\mathfrak{m}$ because we cannot prove that t is the first constant not in ϕ. For all we know, it isn't. Working around this with Dragnet would be a bit of work, but Free Choice makes it easy.

(Case 2:) $\phi t/x$ is not false on at any t-variant of $\mathfrak{m}$. So it's true on all t-variants. Since $(\phi\rightarrow\psi)t/x$ is also true on all t-variants, then by the definition of truth for $\rightarrow$, $\psi t/x$ is true on all t-variants. By the Free Choice theorem, $\forall x\psi$ is true on $\mathfrak{m}$, and thus, by the definition of truth for $\rightarrow$, $\forall x\phi\rightarrow\forall x\psi$ is true on $\mathfrak{m}$.

Nothing particular was assumed about $\mathfrak{m}$. Any model that makes $\forall x(\phi\rightarrow\psi)$ true also makes $\forall x\phi\rightarrow\forall x\psi$ true. Therefore, $\forall x(\phi\rightarrow\psi) \models \forall x\phi\rightarrow\forall x\psi$. ■

4.4 Exercises

4.4.1 Formulas, Order, and Subformulas

Which of the following are *formulas*? For those that are formulas, what is their order? How many subformulas does each have?

1. $\forall x(H'x\rightarrow G'x)$
2. $\forall x(H'x\rightarrow G''x)$
3. $\forall x(H'x\rightarrow G'_7x)$
4. $\forall x\forall z(H'x\rightarrow G''xy)$
5. $\forall x\forall y(H'x\rightarrow G''xy)$
6. $\forall x\forall z(H'\mathrm{a}\rightarrow G''xy)$
7. $\forall x\forall z(Hx\rightarrow G''xy)$
8. $\forall x\forall z(W'x\rightarrow G''xy)$
9. $\forall x\forall z(H'x\rightarrow G''xy)$
10. $\forall x_9\forall z(H'x_9\rightarrow G''xy)$

11. $\sim\forall x\forall y(H'x\rightarrow G''xy)$
12. $\forall \mathrm{b}\forall y(H'\mathrm{b}\rightarrow G''xy)$
13. $\forall x(\forall xH'x\rightarrow G''xy)$
14. $\forall x(\forall zH'x\rightarrow G''xy)$
15. $\forall y(\forall xH'x\rightarrow \forall xG''xy)$

4.4.2 Sentences and Order

For each of the following say whether it is an official sentence, an unofficial sentence, an official formula but not a sentence, an unofficial formula but not a sentence or none of the above. If it is a formula or sentence (official or unofficial) say what its order is and how many sub*formulas* it has.

1. $\forall x\forall y(H'x\rightarrow G''xy)$
2. $\forall x\exists z(H'x\rightarrow G''xy)$
3. $\forall x\forall y(Hx\wedge Gxy\wedge Ky)$
4. $\forall x\forall y(Hx\rightarrow Gxy\rightarrow Ky)$
5. $\forall x\forall y(\sim Hx\wedge Gzy\wedge Ky)$
6. $\forall x\forall y(Hx\wedge(Gzy\wedge Ky))$
7. $\forall x\forall y(H'x\wedge(G''zy\wedge K''y))$
8. $\forall x\forall yHx\wedge(Gzy\wedge Ky)$
9. $\forall x\exists z\forall y(Hx\wedge Gxy\wedge Ky)$
10. $\forall x\exists y\forall z(Hx\wedge\forall yGxy\wedge Ky)$

4.4.3 Truth in a Model

Give the truth value of each of the following sentences on both of the models found in figure 4.2 (on the next page).

1. $\forall x\forall y((Ax\wedge By)\rightarrow Axy)$
2. $\forall x\forall y((Cx\wedge Dy)\rightarrow Dxy)$
3. $\forall x((Cx\wedge Ex)\rightarrow Cx\mathrm{a})$
4. $\forall x\forall y(Cxy\rightarrow Axy)$
5. $\forall xCx\rightarrow\forall yDy$
6. $\forall z\forall w((Gz\wedge Gw)\rightarrow\sim Gzw)$
7. $\forall z\forall w\forall x(Cxzw\rightarrow(Cx\leftrightarrow Cw))$
8. $\forall x(Ax\rightarrow\exists y(Cy\wedge Byx))$
9. $\exists y\exists x(Cxy\wedge Dxy)$
10. $\exists y\exists x(Exy\wedge Gxy)$
11. $\forall x(Gx\rightarrow\forall w(Axw\rightarrow(Aw\vee Cw)))$
12. $\exists x(Cx\rightarrow\forall yCy)$
13. $\forall z\forall w\forall x(Dxzw\rightarrow Axw)$
14. $\forall y(Ay\rightarrow\exists x(Ex\wedge Axy))$
15. $\forall x\forall y(Gxy\rightarrow Gyx)$

4.4.4 Quantificational Truth Problems

For each sentence below say whether it's a quantificational truth. If so, prove it. If not, show it by giving a model $\mathfrak{m}$ that makes it false.

	Symbol	Model	
		Pos Int	**States**
Universe:		The set of positive integers	The set of US states (2024)
Sent. Let.:	A	true	false
	B	true	false
	C	false	true
	D	true	false
	E	true	false
	G	false	true
Constants:	a	1	Louisiana
	b	9	Maine
	c	72	Georgia
	d	3	Nebraska
	e	1	New Mexico
	f	2	Texas
1-place:	A'	all pos int	Midwestern
	B'	empty set	name with > 5 letters
	C'	even	Coastal
	D'	odd	one of original 13
	E'	prime	{Ohio}
	G'	multiple of 7	{Ohio, Alabama}
2-place:	A''	first $>$ second	share a border
	B''	are equal	first is north of second
	C''	first $=$ 2 times second	first $>$ second (area)
	D''	sum of them equals 7	first $>$ second (population)
	E''	first $<$ second	first is west of second
	G''	are relatively prime	both coastal, or neither
3-place:	A'''	all equal	all same population
	B'''	first $<$ second $<$ third	first is north of others
	C'''	all odd or all even	first $>$ second $>$ third (area)
	D'''	first $+$ second $=$ third	first $+$ second $>$ third (area)
	E'''	first $\times$ second $=$ third	first is west of the others
	G'''	are all relatively prime	at least two coastal

Figure 4.2: Two QL models

1. $\forall x \forall y Hxy \rightarrow \forall y \forall x Hxy$
2. $\exists x \exists y Hxy \rightarrow \exists y \exists x Hxy$
3. $\forall x \exists y Hxy \rightarrow \exists y \forall x Hxy$
4. $\exists y \forall x Hxy \rightarrow \forall x \exists y Hxy$
5. $\forall x(Ax \rightarrow \exists y(Hxy \wedge By))$
6. $\exists y(Ay \wedge \forall z(Bz \rightarrow Hyz))$
7. $\forall x(\exists y Hxy \wedge \sim Hxx \wedge \forall y \forall z((Hxy \wedge Hyz) \rightarrow Hxz))$

4.4.5 Preliminary Dragnet Practice Problems

1. If ϕ is $\forall z_3(Gz_1z_3 \wedge \exists x Bxz_3y_2) \rightarrow (A \rightarrow \forall y_2 Dz_3y_2)$, what is
   (a) $\phi \mathrm{a}/z_3$
   (b) $\phi \mathrm{a}/y_2$
   (c) $\phi z_3/x$
   (d) $\phi \mathrm{a}/x$
2. If $\phi x/y$ is Dxx, can you determine what ϕ is? If so, what is it? If not, why not?

4.4.6 Dragnet Practice

For each of the following statements, show whether it is true or false. Use Dragnet or Free Choice whenever it is helpful to do so.

Each of θ, ψ, and ϕ is a formula of QL whose only free variable is x, and none contain the variables y, z or w.

1. $\forall w(\phi w/x \wedge \psi w/x) \models \forall x \phi \rightarrow \forall x \psi$
2. $\forall x \phi \rightarrow \forall x \psi \models \forall y \phi y/x \rightarrow \forall z \psi z/x$
3. $\forall x \phi \rightarrow \forall x \psi \models \exists x(\phi \wedge \psi)$
4. $\forall x(\phi \rightarrow \psi) \models \forall y \phi y/x \rightarrow \forall z \psi z/x$
5. $\forall y \phi y/x \rightarrow \forall z \psi z/x \models \forall x(\phi \rightarrow \psi)$
6. $\forall y \phi y/x \rightarrow \forall z \psi z/x$, $\exists x(\phi \wedge \psi) \models \forall x(\phi \rightarrow \psi)$
7. If $\models \forall x(\phi \rightarrow \psi)$, then $\models \forall x((\phi \wedge \theta) \rightarrow (\psi \wedge \theta))$
8. $\exists y(\phi y/x \wedge \sim \psi y/x) \models \forall y(\phi y/x \rightarrow \sim \psi y/x)$
9. $\sim \forall x \phi \models \exists x \sim \phi$
10. $\models \forall x(\phi \rightarrow \psi) \vee \forall y(\phi y/x \rightarrow \sim \psi y/x)$

Chapter 5
Translations

5.1 SL Applications

Sentences in English have meanings. Declarative sentences say something about the world, and in virtue of their various meanings they are either true or false. By contrast, the sentences of SL and QL are, in themselves, meaningless strings of symbols. Any meaning ascribed to SL and QL sentences comes from an assigned model.

The models in Chapter 2 do not distinguish between different sentences of the same truth value, because for the purposes of assessing logical truth in SL truth value is all that matters. But if we want to translate English into SL we must specify the models less directly: by associating English sentences with sentence letters.

5.1.1 Using SL Models for Translation

Let's say we have some declarative sentences of English that we would like to translate and model in SL. To translate, we first look for the smallest declarative clauses, i.e. the 'atomic' parts, and we associate each with a sentence letter. We decide whether each of these atoms is true or false and then define a model $\mathfrak{m}$ that makes the appropriate assignment to the corresponding sentence letter. If truth values are assigned according to what we know about reality, then $\mathfrak{m}$ is a 'model' of reality, or at least that part of reality described by the English sentences. Alternatively, we can assign truth values so that they do not match reality, in which case $\mathfrak{m}$ is a model of a non-actual state of affairs. Finally, we can assign values to sentences about which the facts are unknown, in which case $\mathfrak{m}$ is a model of a hypothetical state of affairs.

English sentences are not all equally suitable for translation to QL. Ideal English sentences are unambiguous, not vague, and do not have a truth value that changes over time. More practically, we can also model sentences that are unambiguous in the appropriate context, specific enough to have determinate truth values, and whose truth values do not change during some period of interest. One source of well-behaved sentences is mathematics. For other sentences we can usually supply enough context to meet our criteria.

To illustrate, we might say that 'Texas is a US state,' is a true sentence of English. We mean that Texas is a US state at the time the sentence is uttered. By convention, English sentences in the present tense are assessed relative to the time of utterance.

But without such context matters aren't so clear. The sentence is false of Texas in 1844 and true of Texas now, so its truth value has varied over time. The sentence 'Texas was a US state at noon CDT on May 1, 1983,' is less ambiguous, and does not require any context-sensitive judgment about time to assess.

In practice, however, we make background assumptions about what is relevant, who is being discussed, the time of assessment, and so on. E.g., 'Washington was elected President,' has an obvious and natural interpretation, one that's distinct from 'Joan Washington was elected President of the Montrose Dog Walkers Association.'

Note that we speak of translating *sentences* and not, e.g., propositions. There is a controversy in the philosophy of logic over which objects are the most fundamental bearers of truth value: sentences, propositions, judgments, statements, etc. Without weighing in on that debate, we prefer sentences over the alternatives for a logic textbook. We argued in Chapter 1 that there is no uncontroversial definition of a sentence of English. Even so, there is less dispute about what a sentence is than what propositions, judgments, or statements are.

There is difference between sentence tokens and sentence types. We prefer not to deal with the difficulties associated with that distinction in this text. We assume that all sentence tokens of a type have the same truth value, but that isn't true in general.[1]

5.1.2 SL Translation Keys

We use a *translation key* to associate atomic English sentences with SL sentence letters. A translation key is different from a model, but it determines which sentence letter assignments a SL model has to make. A complete model also needs the truth value of each sentence, which isn't supplied by the translation key. Once we've mapped atomic sentences of English to sentence letters we can use the key to translate complex English sentences into SL. Complex sentences are composed of (i) atomic sentences and (ii) logical connectives of English.

When is a sentence of English simple enough to be considered 'atomic'? In putting together a key we should capture as much of the logical structure of the English as is practical and useful. All else being equal, translation keys that hide logical structure in a sentence letter are deficient. For example, the following is not ordinarily a good candidate for sentence letter assignment:

1. Mares eat oats and does eat oats and little lambs eat ivy.

The word 'and' plays a truth-functional role in this sentence, analogous to the role that '$\wedge$' plays in SL. A better translation key would assign each of 'Mares eat oats', 'Does eat oats', and 'Little lambs eat ivy' its own sentence letter. The original sentence of English can then be expressed as a conjunction in SL. In general, the goal of translation to SL is to replace the truth-functional connectives of English sentences with appropriate logical connectives from SL.

[1] See Grandy 1993 for details.

Translation Key:

M: Mares eat oats.
D: Does eat oats.
L: Little lambs eat ivy.

5.1.3 Connectives

Before considering other examples we need to say a little more about connectives. In Chapter 2 (in definition 2.1, on page 15) we listed the "logical connectives" of SL. We've seen the role they play in constructing sentences of SL and in fixing their truth values in a model. But we haven't said anything about what connectives are in general.

The essential idea of a sentential *connective*—whether a "logical" one in some formal language or an expression in a natural language like English—is that it's a part of language that can be used to combine one or more sentences of the language into a new sentence.[2] We say that one connective is within the *scope* of a second connective iff the first connective is within a sentence directly connected with the second connective. The *main connective* is the one connective of the sentence that's not within the scope of any other connectives.

A connective is *truth-functional* iff the truth value of every new sentence formed by that connective depends solely on the truth value of the constituent sentences. (If we're talking about a connective in a formal language like SL, we mean the truth value of a sentence *in a model*.) Not all connectives of English are truth-functional. In particular, there are modal connectives—such as 'necessarily' and 'possibly'—that defy truth-functional characterization. For example, the sentence

2. Necessarily, bachelors are unmarried.

seems plausibly true. If we replace 'bachelors are unmarried' with another true claim, however, the result is not so plausible:

3. Necessarily, there is life on earth.

Both 'bachelors are unmarried' and 'There is life on earth' are true. But it is easier to imagine all life on earth perishing than for bachelors to be married. That there is life on earth is contingent. The truth of a sentence with a modal main connective depends on more than the truth value of the part of the sentence governed by the modal connective.

By contrast, the five logical connectives of SL are all truth-functional. This disparity means that SL is not capable of capturing all of the logical structure of English. The best it can do is to approximate certain features of English. Some formal languages have the resources to capture the logical structure of modal claims. We introduce

[2] Other connectives play a subtler role, as with the quantifiers of QL. In QL quantifiers can be used to combine predicates to make a more complex predicate.

one such language in Chapter 9. We hasten to add, however, that SL has sufficient resources to render many English sentences without significantly distorting their basic meaning.

5.1.4 Conjunctions and Negations

A logical connective of SL is a suitable translation of a (truth-functional) connective of English iff they express the same truth function (see section 2.2.2, on page 25). That is, iff the new sentences formed by those connectives share the same truth value whenever the constituent sentences that make them up share the same truth values. We've already mentioned one such pair of corresponding connectives: 'and' and '$\wedge$'. The sentence

4. The sun is shining *and* the birds are chirping.

is true iff the following two sentences are also true:

5. The sun is shining.
6. The birds are chirping.

The same holds of other conjunctions in English. The SL connective '$\wedge$' is a good translation of 'and' most of the time. Sometimes the structure of English conjunctions is more hidden, as in the following example:

7. Nathanael Green *and* 'Light Horse Harry' Lee were officers in the Continental Army.

A translation key should not break the sentence at the word 'and' and give the LHS—'Nathanael Green'—its own sentence letter. The LHS is not a declarative sentence. It's clear that 7 expresses the conjunction of the following two sentences:

8. Nathanael Green was an officer in the Continental Army.
9. 'Light Horse Harry' Lee was an officer in the Continental Army.

It is appropriate to map each of these latter sentences to a sentence letter—say, N and L, respectively—and express 'Nathanael Green and 'Light Horse Harry' Lee were officers in the Continental Army' as a conjunction of these sentence letters—e.g., $(N \wedge L)$.

Other words of English also play the role of conjunction. Consider:

10. Mary wants to drive her mother's car, *but* she is *not* old enough.

The word 'but' connects two independent claims. The sentence as a whole is true iff the LHS and the RHS are each true. So 'but' plays the same truth-functional role as '$\wedge$' in SL.

The word ‘not’ is another connective; it corresponds to the ‘$\sim$’ of SL. The words “not” and “no” are typically translated with $\sim$.

Negations in English are not always so obvious. “Innumerable” is not numerable, “unmistakable” is not mistakable, “never” is not ever, “immoral” is not moral, “clueless” is to have no clue, and so on. Yet while the prefix “in-” is often negation, as in “inevitable”, it is also often a direction, as in “influx”. The latter case is not one of negation.

Notice that in the RHS of 10, context indicates that ‘she’ is Mary, and that mention of her being not ‘old enough’ is about the fact that she is not old enough *to drive her mother’s car*. English is, in this respect, more efficient than SL. To translate this sentence into SL we need something like:

Translation Key:

A: Mary wants to drive her mother’s car.
O: Mary is old enough to drive her mother’s car.

We take out the ‘not’ because we can represent that with ‘$\sim$’. We render the original sentence in SL as $(A \wedge \sim O)$. This is indistinguishable from the result of translating

11. Mary wants to drive her mother’s car *and* she is not old enough to drive her mother’s car.

In English we typically say ‘Φ but Θ’ instead of ‘Φ and Θ’ to signal to listeners that they are about to hear something surprising. They may expect that Φ and Θ are an unlikely pairing. This difference from the word ‘and’ cannot be captured in SL—it is lost in translation. SL translations are approximations of English, and as such they can’t always preserve the original meaning.

The word ‘and’ can do more than serve as a conjunction. Sometimes the order of the conjuncts affects meaning. Compare:

12. John took off his clothes and went to bed.

13. John went to bed and took off his clothes.

The order of the conjuncts suggests that John performed these actions in a different order. Yet if we translate 12 and 13 into SL the results are truth-functionally equivalent. We think this equivalence is defensible, because 12 and 13 don’t explicitly describe the order of events. We tend to read order into them because it’s usual in conversation to recount events in chronological order. This is not a requirement, however. There is nothing contradictory about the following:

14. John went to bed and took off his clothes, but I don’t know in which order.

English	SL
Φ and Θ	$\phi \wedge \theta$
both Φ and Θ	$\phi \wedge \theta$
Φ, but Θ	$\phi \wedge \theta$
Φ, but not Θ	$\phi \wedge \sim\theta$
not Φ, but Θ	$\sim\phi \wedge \theta$

Table 5.1: Translations for Common English Conjunctions

5.1.5 Disjunctions

Another connective of English is the word 'or':

15. *Either* the tigers will get us *or* the lions will.

The sentence as a whole is true iff a clause on either side of the 'or' is true. This matches the truth-function of the '$\vee$' connective in SL.

Translation Key:

I: The tigers will get us.
L: The lions will get us.

The resulting translation: $(I \vee L)$. We again use context to fill out the sentence 'the lions will [get us]'.

It might not seem quite right to translate 'or' as '$\vee$'. In SL $(\phi \vee \theta)$ is true when both ϕ and θ are true on some model. But sometimes in English when we say 'Φ or Θ' we mean that one or the other of Φ and Θ is true—but not both. A disjunction in which both disjuncts can be true is *inclusive*, while a disjunction in which only one disjunct can be true is *exclusive*. Should 'Φ or Θ' be considered exclusive? Is it incorrect to translate it as $(\phi \vee \theta)$?

These are complicated questions that we cannot fully answer here. We believe, however, that 'or' is typically inclusive in English and so best translated as '$\vee$'. Only in certain conversational contexts is the exclusive disjunction clearly intended. We motivate our stance with a few examples.[3]

One case in which 'or' sounds exclusive involves disjuncts that cannot possibly both be true. I might say "Lydia either took a boat or a plane". It's not physically possible for Lydia to travel by distinct modes of travel at the same time. However,

[3]This discussion borrows from Smith 2012: 117–22.

English	**SL**
Φ or Θ	$\phi \vee \theta$
either Φ or Θ	$\phi \vee \theta$
Φ or Θ, but not both	$(\phi \vee \theta) \wedge \sim(\phi \wedge \theta)$

Table 5.2: Translations for Common English Disjunctions

we must distinguish the meaning of the sentence from what we otherwise know to be possible. We know that it's impossible to travel by boat and by plane at the same time. It's tempting to assimilate that fact into the sentence's meaning, but that leap is unjustified. Someone could understand "Lydia either took a boat or a plane" and agree that it's true, but also think that both disjuncts are true. Perhaps Lydia's journey was disjointed and involved both modes of travel. To conclude that this disjunction is exclusive requires more information than the sentence itself conveys.

There are two cases in which it's more plausible that 'or' is exclusive. The first case involves commands or rules. If you are at a fancy restaurant and a waiter says, "You may have the soup or a salad", it's typically understood that you may have one or the other, but not both.

The second case involves elliptical clauses. The phrase 'Φ or Θ' can be short for 'Φ or Θ, but not both'. If so, then the intended message is exclusive. But this isn't a case in which the 'or' itself expresses exclusive disjunction. The 'or' plus the tacit phrase 'but not both' together express exclusive disjunction.

Special cases like these, which are rare but salient, might lead you to think that disjunctions in English are sometimes exclusive. However, in these cases there are context clues indicating their exclusive character, separate from the meaning of the relevant sentences. We know as a matter of cultural familiarity that restaurants don't offer *both* soup *and* salad unless we pay extra. When intergalactic visitors discover Earth and visit our restaurants, we recommend that waiters make the exclusivity of the disjunction explicit. Otherwise there could be an interstellar diplomatic incident! They can even use SL to do so:

16. The intergalactic visitor may have the soup *or* a salad, *but not* both.

This sentence has several truth-functional connectives, so we must be careful to understand which connectives govern which. The 'but' governs everything else, so the whole sentence is a conjunction. The visitor may have the soup or a salad, *and* he may not have both the soup and a salad. The translation key:

Translation Key:

P: The visiting intergalactic visitor may have the soup.
D: The visiting intergalactic visitor may have a salad.

First we translate the LHS of the 'but': $(P \vee D)$. If we translate the RHS without the 'not'—i.e., 'the intergalactic visitor may have both soup and a salad'—we get: $(P \wedge D)$. We negate the result to account for the 'not': $\sim(P \wedge D)$. Putting it all together, the resulting translation is: $((P \vee D) \wedge \sim(P \wedge D))$.

5.1.6 Conditionals and Biconditionals

Another connective of English is the *conditional*. Conditionals are expressible in many ways, but perhaps the most distinctive way is 'if ... then ....'

17. If George Washington crosses the Delaware River then the Hessians will be defeated.

Is the conditional of English a truth-functional connective? Philosophers have been arguing over how to answer this question since the ancient debate between Diodorus Cronus and Philo of Megara over 2000 years ago. SL has only truth-functional connectives, so if the conditional is not truth-functional then '$\rightarrow$' cannot fully capture its meaning. We can show, however, that '$\rightarrow$' is the best truth-functional representation of the conditional. As we discuss translations the reader will get a sense of how well the model fits English.

Let's consider a few examples. An SL sentence of the form $(\phi \rightarrow \theta)$ is false on a model $\mathfrak{m}$ iff $\mathfrak{m}$ makes ϕ true and θ false. That gives us the correct assessment of the English sentence:

18. If $2 + 2 = 4$ then $4 + 4 = 6$.

On all other combinations of truth values for ϕ and θ, SL sentences of the form $(\phi \rightarrow \theta)$ are true. Consider a case in which both the LHS and the RHS of a conditional are false. Some such conditionals are true:

19. If George Washington landed on the moon then George Washington landed on the moon.

This sentence is not merely true. It's a logical truth. Other instances are more dubious:

20. If Soviet cosmonauts landed on the moon in 1968, then George Washington was never elected President.

No truth-functional connective allows us to distinguish the last two examples. In both cases we can only refer to the truth values of each side: the LHS and the RHS are false. In order to admit 19 as true, we must also admit 20. This is one price we must pay for truth-functional conditionals.

Here is a sentence in which both the LHS and the RHS are true:

English	**SL**
if Φ, then Θ	$\phi \rightarrow \theta$
Φ only if Θ	$\phi \rightarrow \theta$
Φ if Θ	$\theta \rightarrow \phi$
Φ provided that Θ	$\theta \rightarrow \phi$
provided Φ, Θ	$\phi \rightarrow \theta$
Φ assuming that Θ	$\theta \rightarrow \phi$
assuming Φ, Θ	$\phi \rightarrow \theta$
for Φ, it's necessary that Θ	$\phi \rightarrow \theta$
for Φ, it's sufficient that Θ	$\theta \rightarrow \phi$
Φ if and only if Θ	$\phi \leftrightarrow \theta$
for Φ it's necessary and sufficient that Θ	$\phi \leftrightarrow \theta$
Φ just when Θ	$\phi \leftrightarrow \theta$
Φ just in case Θ	$\phi \leftrightarrow \theta$

Table 5.3: Translations for Common English Conditionals and Biconditionals

21. If George Washington crossed the Delaware River then George Washington crossed the Delaware River.

As with 19 this is not merely true, it's a logical truth. Consider an SL model that makes ϕ false and θ true, and hence makes $(\phi \rightarrow \theta)$ true. We can replace the LHS of 21 with a false conjunction and the result is still true:

22. If George Washington crossed the Delaware River and Thomas Jefferson invented bifocals, then George Washington crossed the Delaware River.

The true inventor of bifocals was probably Benjamin Franklin. Even though the conjunction on the LHS is false, the sentence is still a logical truth. The corresponding SL translation is TFT: $((G \wedge B) \rightarrow G)$. So, while the '$\rightarrow$' is not a perfect match for the English language conditional, it is the best truth-functional translation.

Biconditionals in English are closely related to conditionals, so they're subject to some of the same worries. Nevertheless, we treat them as truth-functional connectives for the purpose of translating them into SL:

23. Ruth may play outside if and only if she cleans her room.

Translation Key:

H: Ruth may play outside.
C: Ruth cleans her room.

With this key we translate the sentence as: $(H \leftrightarrow C)$. We use '$\leftrightarrow$' to translate English biconditionals.

Sometimes the word 'if' is used to express a biconditional:

24. Ruth may play outside if she cleans her room.

But what if she doesn't clean her room? Presumably she will not be allowed to play outside. Yet this is not the literal meaning of the word 'if'. We use context to supply the implicit 'but not otherwise' to the sentence. It is well-known that parents use punishments and rewards to shape child behavior, so we rightly interpret 24 as shorthand.

In general it is our policy is to translate what each sentence says literally. If it is important for an argument to articulate what is being added by context, that should be specified explicitly.

5.1.7 Further Examples

Let's look at a few more translations.

Example 5.1.

25. Either Lydia flew or both Jackson and Helen took off early.

The first step is to identify the main connective of the sentence: 'either ... or'. We translate 'either ... or' as $\vee$:

26. (Lydia flew) $\vee$ (both Jackson and Helen took off early).

Notice that the word "both" clarifies the logical structure of the sentence. Without it, the sentence is, "Either Lydia flew or Jackson and Helen took off early." This is ambiguous, and could be read as saying that either Lydia or Jackson flew.

Next we identify and translate the main connectives in the connected clauses. There are no connectives in 'Lydia flew', so there is nothing to do with it. There is one connective in 'both Jackson and Helen took off early', 'both ... and'. We translate this as '$\wedge$':

27. (Lydia flew) $\vee$ ((Jackson took off early) $\wedge$ (Helen took off early)).

Translation Key:

N: Lydia flew.

J: Jackson took off early.

H: Helen took off early.

28. $N \vee (J \wedge H)$

Example 5.2.

29. If Lydia got the job and Jackson didn't, then Lydia will take off tomorrow and Helen will have to come in.

We identify the main connective, 'if ... then'. This is translated as '$\rightarrow$':

30. (Lydia got the job and Jackson didn't) $\rightarrow$ (Lydia will take off tomorrow and Helen will have to come in).

Next we look at the LHS of the conditional, 'Lydia got the job and Jackson didn't'. The main (and only) connective of this sentence is 'and', which we translate as '$\wedge$':

31. ((Lydia got the job) $\wedge$ (Jackson didn't get the job)) $\rightarrow$ (Lydia will take off tomorrow and Helen will have to come in).

The right-hand conjunct 'Jackson didn't' is an elliptical clause. Lydia got the job and Jackson didn't *get the job*. We make this tacit phrase explicit by putting it in brackets. The clause 'Jackson didn't' has a negation in it. It combines 'not' with 'Jackson got the job'. We finish the translation by translating the conjunction on the RHS of the sentence:

32. ((Lydia got the job) $\wedge$ ($\sim$ (Jackson got the job))) $\rightarrow$ ((Lydia will take off tomorrow) $\wedge$ (Helen will have to come in)).

There are no more connectives, so we are ready to construct a translation key.

Translation Key:

N: Lydia got the job.

J: Jackson got the job.

I: Lydia will take off tomorrow.

H: Helen will have to come in late.

33. $(N \wedge \sim J) \rightarrow (I \wedge H)$

Example 5.3. Some connectives in English do not correspond to any single SL connective. However, we can express any truth-function using some combination of the SL connectives that we *do* have. For example:

English	**SL**
not Φ	$\sim\phi$
it's not the case that Φ	$\sim\phi$
Φ unless Θ	$\sim\theta \rightarrow \phi$
Φ unless Θ	$\theta \vee \phi$
unless Φ, Θ	$\sim\phi \rightarrow \theta$
Φ if not Θ	$\sim\theta \rightarrow \phi$
neither Φ nor Θ	$\sim(\phi \vee \theta)$
	$\sim\phi \wedge \sim\theta$
not both Φ and Θ	$\sim(\phi \wedge \theta)$
	$\sim\phi \vee \sim\theta$

Table 5.4: Translations for Common English Negations and Complex Connectives

34. Neither Lydia nor Helen were late.

The connective is 'neither ... nor', and it joins 'Lydia [was late]' and 'Helen [was] late'. There is no single SL connective that has the same truth-function as 'neither ... nor'.[4] Recall theorem 2.80 (on page 50), which says that any truth-functional connective can be expressed in SL. It follows that there is some combination of SL connectives we can use to translate 'neither Φ nor Θ'. There are two equivalent translations: $\sim(\phi \vee \theta)$ and $\sim\phi \wedge \sim\theta$. So we have:

35. $\sim$ ((Lydia was late) $\vee$ (Helen was late)).

With the key:

Translation Scheme:

N: Lydia was late.
H: Helen was late.

The final translation of sentence 34 is:

36. $\sim(N \vee H)$

[4] But recall the logical connective NOR, discussed at the end of section 2.6.2 (on page 47). NOR correctly translates 'neither ... nor', but isn't part of SL.

The tables in this chapter should be thought of as rough-and-ready guides. Although many particular uses of 'and' express conjunction, not all do. Sometimes 'and' doesn't function as a connective at all, e.g. as in 'it will be years and years before the trees bear fruit' (Smith 2012: 107). Other times 'and' functions as a connective, but expresses a conditional instead of a conjunction, e.g. 'study hard, and you will pass the exam' (Smith 2012: 107).

To translate a connective from English appropriately you must first determine what is being expressed, drawing on context as necessary to resolve ambiguities. For example, "Maria and Paul are married," can convey the conjunction "Maria is married and Paul is married," or it can express "Maria and Paul are married to each other," in which case the 'and' is not a conjunction, but is serving a different logical purpose to be discussed in the next section.

5.2 QL Applications

5.2.1 Constants and Predicates

Let's say we want to translate the following sentence into *SL*:

1. Mary is happy, smart, adorable, and a child.

The only connective we can translate is the 'and'. The translation key:

Translation Key:

H: Mary is happy.
R: Mary is smart.
A: Mary is adorable.
C: Mary is a child.

The SL result is: $(H \wedge R \wedge A \wedge C)$. This translation might work for certain purposes, but the conjuncts have no logical connection with each other. Nothing in QL indicates that each conjunct is about the same person.

In QL we can express that shared logical connection with constants and predicates. Translation keys of QL connect nouns to constants and English predicates to QL predicates. Let's use the following QL key to translate 1:

Translation Key:

m: Mary
Ht: t a child.
Rt: t is smart.
At: t is adorable.
Ct: t is a child.

The result is: $(H\text{m} \wedge R\text{m} \wedge A\text{m} \wedge C\text{m})$. This translation allows us to see that we are predicating several things about the same person. On to another example:

2. Ronnie and Demaryius are athletic, but Peyton isn't.

We *could* translate this as a conjunction with three conjuncts in SL. However, this would not make clear that the same predicate either applies, or doesn't, to each of the three. Instead, let's use the following QL key:

Translation Key:

r: Ronnie
d: Demaryius
p: Peyton
At: t is athletic.

The result: $(A\text{r} \wedge A\text{d} \wedge \sim A\text{p})$. We see the common predicate in each conjunct.

The translation keys here resemble the QL models we provided earlier.[5] One difference is that we don't have a domain assigned in either of the QL keys above. We could add a domain to each of the above, and interpret the constant and predicate lines of the key as making assignments from the domain. Hence, we can usually treat informal model assignments as translation keys.

5.2.2 Quantifiers

The quantifier '$\forall$' corresponds to the English phrases 'all' or 'every'. Let's say we want to claim that every member of some set is also a member of some other set:

3. All dogs are furry.

There are two sets: the set of all dogs and the set of all furry things. Sentence 3 effectively claims that the set of dogs is a subset of the set of furry things. How do we translate this into QL? The word 'all' typically calls for a universal quantifier.

It isn't obvious from the sentence itself, but whenever we want to make a claim of the form 'All Φ are Θ', we nearly always want to translate it as: $\forall\alpha(\phi \rightarrow \theta)$. That is, with a '$\forall$' governing an '$\rightarrow$'. This makes more sense if we paraphrase 3 in MathEnglish as: 'For all x, if x is a dog then x is furry.'

For the rest of the translations in this section we use the following model description as our translation key:

Animals model:

$Animals(U)$: All animals.

[5] See table 4.1 in Chapter 4.

$Animals(A')$: is a mammal.
$Animals(C')$: is a cat.
$Animals(D')$: is a dog.
$Animals(E')$: is energetic.
$Animals(H')$: is a happy.
$Animals(R')$: is furry.
$Animals(A'')$: is smarter than.

We translate 3 as: $\forall x(Dx \rightarrow Rx)$.

Other English words can be translated with the universal quantifier. The following uses the word 'no' to make a universal claim:

4. No dogs are furry.

Again, we can think of this as a claim about two sets: the set of dogs and the set of furry things. This sentence is tantamount to a claim that these sets are disjoint, i.e., that nothing is a member of both. We usually translate such claims into the following form in QL: $\forall\alpha(\phi \rightarrow \sim\theta)$. To understand why, consider the following MathEnglish paraphrase: 'For all x, if x is a dog then x is not furry.' The resulting translation is: $\forall x(Dx \rightarrow \sim Rx)$.

The word 'only' can also be used for universal claims:

5. Only dogs are furry.

This is translated in the same way as sentence 3, except that we reverse the order of the LHS and the RHS of the conditional governed by the '$\forall$'. It's equivalent to the claim that 'All furry things are dogs.' So: $\forall x(Rx \rightarrow Dx)$.

The QL symbol '$\exists$' corresponds to the English phrases 'there exists', 'there is', or 'some'. Consider the following existential sentences.

6. Some dogs are furry.
7. Some dogs are not furry.

If we again think of the set of dogs and the set of furry things, 6 is a claim that there is at least one element that is a member of each set. Notice that we are interpreting 6 as a claim about at least one object. In English we typically think that 6 is a claim about at least two dogs. For now we ignore certain plural/singular distinctions in the way we interpret existential claims. For our purposes, 'some' means 'at least one'. We handle 'some' in a more satisfactory way when we add to QL in Chapter 9.

For now, we translate sentences like 6 into the form: $\exists\alpha(\phi \wedge \theta)$. So, for 6 itself: $\exists x(Dx \wedge Rx)$. And we account for the 'not' in sentence 7 as follows: $\exists x(Dx \wedge \sim Rx)$.

We can also translate sentence 4, 'No dogs are furry', as an existential governed by a negation. We could translate 'There is a furry dog' as: $\exists x(Rx \wedge Dx)$. We can

negate the result to capture the meaning of 4: $\sim\exists x(Rx \wedge Dx)$. In fact, this latter translation is logically equivalent to the one we gave earlier: $\forall x(Dx \rightarrow \sim Rx)$. To see this, join these two sentences together with a biconditional, '$\leftrightarrow$', and prove that the result is QT.

Let's translate more complicated sentences into QL:

8. All happy dogs are furry and energetic.

We want to translate this as a universal quantifier governing a conditional, but we must be careful to translate each side of the conditional correctly. We paraphrase 8 in MathEnglish as: For all x, if (x is happy and x is a dog) then (x is furry and x is energetic). So we can consider each side of the conditional as a conjunction: $\forall x((Hx \wedge Dx) \rightarrow (Rx \wedge Ex))$.

9. All cats and dogs are mammals.

This sentence is also going to be translated as a universal quantifier governing a conditional, but the word 'and' can be tricky. Here 'and' seems to conjoin predicates, not sentences. We may be tempted to translate 9 as: $\forall x((Cx \wedge Dx) \rightarrow Ax)$. But this QL sentence can be translated into MathEnglish as: 'For every x, if (x is a cat and x is a dog) then x is a mammal.' But that's silly. Unless mad scientists are involved, nothing is both a cat and a dog. Instead, we should translate the 'and' in 9 as a '$\vee$': $\forall x((Cx \vee Dx) \rightarrow Ax)$. Let's translate this QL sentence into MathEnglish: 'For all x, if (x is a cat or x is a dog), then x is a mammal.' Take a moment to see how this better expresses the meaning of 9.

Consider a sentence with multiple quantifiers:

10. All dogs are smarter than all cats.

There are two instances of the word 'all', so we use two universal quantifiers in the translation. Consider a paraphrase into MathEnglish: 'For every x, if x is a dog then (for all y, if y is a cat then x is smarter than y).' So we translate 10 as: $\forall x(Dx \rightarrow \forall y(Cy \rightarrow Axy))$.

We have not said much about the role of domains in translations. Because the point of translations, other than the sheer joy of doing it, is to evaluate arguments, it is appropriate to choose a domain suitable for the arguments in question. Often a suitable choice of domain simplifies the translations. If we are translating arguments that mention both dogs and natural numbers, we need to include both in the domain and to have a predicate for each. If the arguments deal only with dogs then we can take the domain to be dogs. You may be able to do without a predicate for 'dog', depending on the nature of the argument.

As we observed before many English sentences are ambiguous. One systematic ambiguity is in sentences of the form 'All As are not Bs', which can either mean that it is not true that 'All As are Bs', or that all 'As are not-Bs'. Context or content

usually indicate what is meant: 'All sheep are not good pets' likely has the first meaning: $\sim\forall x(Sx \rightarrow (Gx \wedge Px))$. But 'All sharks are not good pets' has the second: $\forall x(Sx \rightarrow \sim(Gx \wedge Px))$. One of the values of formalization is that we can clearly and unambiguously express the structure of both.

It is important to distinguish ambiguous sentences—English sentences that have more than one meaning—from sentences for which there is more than one good translation, but which are equivalent. We see this with 'No sharks are good pets': $\forall x(Sx \rightarrow \sim(Gx \wedge Px))$ and $\sim\exists x(Sx \wedge (Gx \wedge Px))$ are equally good (and equivalent) translations.

5.3 Exercises

5.3.1 SL to English Translations

Given the following translation key, translate the following SL sentences into English.

Translation Key:

C: Cindy the Capybara is a picky eater.
O: Oscar the Ocelot sleeps all day.
R: Ralph the Rhinoceros goes for a swim.
A: France is east of Spain.

1. $C \rightarrow O$
2. $O \rightarrow C$
3. $\sim(R \rightarrow A)$
4. $\sim R \rightarrow A$
5. $(C \wedge O) \vee \sim(C \leftrightarrow A)$
6. $C \wedge (O \vee \sim(C \leftrightarrow A))$
7. $A \rightarrow \sim(C \wedge R)$
8. $(C \wedge R) \rightarrow \sim A$

5.3.2 English to SL Translations #1

Using some sensible translation key translate the following English sentences into SL.

1. If the sprockets come in on time, then we can fill the order.
2. Only if the sprockets come in on time can we fill the order.
3. Either the order gets filled, or the cogs come in late and the sprockets never show up.
4. It's not the case that the sprockets need to come in for the order to be filled.
5. While filling the order is important, getting the sprockets in is more so.

	Symbol	Model Assignment
Universe:		The set of states
Constants:	c	CA
	m	MT
	h	RI
	e	TX
1 place predicates:	P'	Pacific states
	A'	Atlantic states
	G'	Gulf states
	M'	Mountainous states
	C'	Coastal states
2 place predicates:	L''	is larger than (area)
	B''	borders

Figure 5.1: Model for Translations in Section 5.3.4

6. The sprockets and cogs are late, but it's still not the case that we can't fill the order on time.
7. Assuming the order gets out on time, the sprockets will fail to arrive only if the cogs are either late or defective.
8. The spork is the least appreciated utensil.
9. They dined on mince, and slices of quince, [which] they ate with a runcible spoon. (1871, Edward Lear, "Owl & Pussy-Cat" in *Nonsense Songs*)
10. You eat with a spork if and only if you eat with a foon.
11. Although Jan will be amused, if you eat with a spork Jill will leave or at least not laugh.
12. If you have a runcible spoon, then you don't need a fork, knife, or spoon.

5.3.3 English to SL Translations #2

Using some sensible translation key translate the following English sentences into SL.

1. If 14-year-olds had the vote, I'd be president. (Evel Knievel)
2. If Miami beats Cornell today and Penn State defeats Michigan State Miami will win the tournament.
3. Should senator Ervin run again, he would be a formidable opponent.

4. Provided, but only provided, that the French Fleet is sailed forthwith for British harbors, His Majesty's Government give their full consent to an armistice for France. (Churchill, June 1940)

5. For the tenability of the thesis that mathematics is logic it is not only sufficient but also necessary that all mathematical expressions be capable of definition on the basis solely of logical ones. (W.V.O. Quine)

5.3.4 Translations

Translate each of the following English sentences into QL sentences about the model $\mathfrak{m}$ given in figure 5.5 (on the previous page).

1. All Pacific states that border a mountainous state are coastal.
2. Some Atlantic state and some mountainous state both share a border with a state that is neither.
3. All states are coastal and mountainous if and only if they are Pacific.
4. All Atlantic states smaller than Montana share a border with Rhode Island.
5. Only mountainous Pacific states are coastal.
6. A Pacific state is mountainous.
7. No state is larger than itself.
8. Every non-mountainous state borders a state that is larger.
9. Some Pacific states are mountainous.
10. All Pacific states are mountainous.
11. All Pacific states are larger than all Atlantic states.
12. No Gulf state is mountainous.
13. All Atlantic states are not mountainous.
14. Some Gulf state is larger than all states that border it.
15. Some Gulf state is an Atlantic state.
16. All states that border a Pacific state are mountainous.
17. Any state that is mountainous is larger than Rhode Island.
18. Any state that is mountainous is larger than all Atlantic states.
19. If any state is mountainous, California is.
20. If any state is mountainous, it is larger than Rhode Island.
21. Any state that has no bordering states is mountainous.
22. All states that are bigger than all mountainous states are coastal.
23. No state is bigger than Montana unless it is coastal.

5.3.5 More Translations

Translate each of the following English sentences into QL sentences.

1. All beavers avoid some kangaroo.
2. All beavers avoid all kangaroos.
3. Some beaver avoids all kangaroos.
4. Every kangaroo is avoided by some beaver.
5. All beavers avoid any kangaroo that frightens them.
6. Some beavers avoid any kangaroo that frightens them.
7. No kangaroo frightens any beaver.
8. No beaver is frightened by any kangaroo.
9. No beaver avoids a kangaroo unless the kangaroo frightens it.
10. Some kangaroo frightens itself.
11. No beaver avoids a kangaroo unless the beaver frightens the kangaroo.
12. Any kangaroo that is frightened of itself is frightened by any beaver.
13. Beavers avoid kangaroos only if they frighten them.
14. Kangaroos that frighten beavers frighten themselves.
15. All kangaroos avoid any kangaroo that avoids them.
16. When a kangaroo frightens a beaver, the beaver avoids it.
17. Beavers only avoid kangaroos.
18. Beavers are frightened of all kangaroos unless they avoid them.
19. Some beavers avoid only kangaroos that frighten them.
20. No beaver that avoids all kangaroos frightens itself.

Chapter 6

Sentential Derivations

6.1 Introduction

We made sense of semantic notions like truth, logical truth, and entailment in previous chapters by the use of SL and QL models. But proving that a sentence of SL or QL is a logical truth, or proving that an entailment holds, is often difficult and involves informal reasoning in MathEnglish. We would like to prove such things more easily and with less reliance on intuitive judgment. Recall from section 1.2.2 that SL and QL are *formal*. As a result, one can determine whether a strings of symbols is a sentence in a purely mechanical way. Are there similarly formal methods for establishing logical truths and entailments?

Yes. In this chapter we define rules for manipulating SL sentences, allowing us to derive other sentences by formal steps. We call a finite sequence of such steps a *derivation*. Every sentence is justified by a rule which permits us to write it down, usually on the basis of previous sentences in the derivation.[1] Here's an example derivation:

1.	$B \wedge C$	*Assume*
2.	B	$\wedge$-*Elim*, 1
3.	C	$\wedge$-*Elim*, 1
4.	$C \wedge B$	$\wedge$-*Intro*, 2,3
5.	$(B \wedge C) \rightarrow (C \wedge B)$	$\rightarrow$-*Intro*, 1–4[2]

Each step contains three parts: a line number, a sentence, and a rule. Derivations are formal because the rules they are constructed with are formal. That is, the rules

[1] The terms 'derivation' and 'proof' (or 'formal proof') are often used interchangeably in other texts. The subfield of logic that studies derivations is even called proof theory. We always use 'derivation' to refer to the sequences of formal steps defined in this chapter and 'proof' to refer to the more informal mathematical proofs written in MathEnglish.

[2] This kind of introduction and elimination rule-based derivation system is called a *natural deduction* system. Natural deduction systems were invented by Stanislaw Jaskowski in 1926, though he did not publish until Jaskowski (1934). Other important developments are due to Gerhard Gentzen (1934). Unlike many systems, ours uses boxes to discharge assumptions. Jaskowski mentions in the 1934 article that he used boxes in 1926, but they aren't part of his 1934 system.

are specified entirely in terms of the shape, or form, of the sentences in the derivation. So there are no rules like:

1. If $\phi \models \psi$ and you have ϕ on a previous line then you may write down ψ.

It can be arbitrarily difficult to determine whether $\phi \models \psi$, so it isn't always obvious whether a given application of this rule is correct. By contrast, formal rules are transparent and easy to check for correctness:

2. If you have $\phi \wedge \psi$ on a previous line then you may write down ψ.

There is no explicit connection between the derivation rules and the semantic notions of truth, logical truth, and entailment. Even so, the rules are intended to give us the same results that were established about these notions in previous chapters. For example, the rules are carefully defined to be truth-preserving.

Truth-preservation: If a rule allows the derivation of sentence ϕ from sentences $\psi_1, \dots, \psi_n$, then $\psi_1, \dots, \psi_n \models \phi$.

Derivations should be correct. That is, any sentence that can be derived without assumptions should be a logical truth, and any sentence derived from other sentences should be a logical consequence of them:

Soundness: If ϕ can be derived from $\psi_1, \dots, \psi_n$, then $\psi_1, \dots, \psi_n \models \phi$.

We want derivations to work in the other direction too. If a sentence is a logical truth, or if some sentence is a logical consequence of some other sentences, there should be a derivation showing as much:

Completeness: If $\psi_1, \dots, \psi_n \models \phi$, then ϕ can be derived from $\psi_1, \dots, \psi_n$.

We prove these two results (and variations of them) for SL (and QL) in a later chapter. For now it is enough to remember that derivations are intended to mirror many of the properties of the proofs of earlier chapters.

It's useful to think of writing a derivation as playing a game or solving a puzzle. Like a game there are rules determining what moves you can make next, or like a puzzle there's a sequence of steps you can discover to make the derivation fit together. Unlike ordinarily puzzles, however, there are many ways they can fit together.

6.2 The Basic System SD

6.2.1 Introduction and Elimination Rules

There are two kinds of rules: basic and shortcut. The basic rules are the minimum rules needed to define derivations. The shortcut rules are not necessary but they make derivations easier and shorter. Anything we can derive with the shortcut rules can be derived from the basic ones alone (see thm. 6.7, on page 148).

The rules for sentences of SL make up *Sentential Derivation System*, or SD. SD consists of the following rules: *Assume*, *Repetition*, and an introduction rule and elimination rule for each of the SL connectives.

We begin by considering the introduction and elimination rules for the conjunction, $\wedge$. If we have the sentences ϕ and ψ in a derivation, then we may write $\phi \wedge \psi$ on a new line, by the $\wedge$ introduction rule. On the other hand, if we have the sentence $\phi \wedge \psi$, then the $\wedge$ elimination rule permits us to write either ϕ or ψ on a new line. Table 6.2 gives all the basic SD rules.

Name	**Given**	**May Add**
Assume		$\mid \phi$
Rep.	ϕ	ϕ
$\rightarrow$*-Elim*	$\theta \rightarrow \psi$, θ	ψ
$\rightarrow$*-Intro*	$\mid \theta$	
	$\mid \vdots$	
	$\mid \psi$	$\theta \rightarrow \psi$, Draw box[3]
$\wedge$*-Elim*	$\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n$	Any one of the conjuncts i.e., θ_i
$\wedge$*-Intro*	θ_1, θ_2, ... θ_n	$\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n$
$\vee$*-Elim*	$\theta_1 \vee \theta_2 \vee ... \vee \theta_n$,	
	$\theta_1 \rightarrow \psi$,	
	$\theta_2 \rightarrow \psi$,	
	$\vdots$	
	$\theta_n \rightarrow \psi$	ψ
$\vee$*-Intro*	θ	$\psi_1 \vee \psi_2 \vee ... \vee \psi_n$, where θ is ψ_i for some i.
$\sim$*-Intro*	$\theta \rightarrow (\psi \wedge \sim\psi)$	$\sim\theta$
$\sim$*-Elim*	$\sim\theta \rightarrow (\psi \wedge \sim\psi)$	θ

Table 6.2: Basic Rules of SD

Continued next Page

[3]When using the rule $\rightarrow$-Intro, you must draw a box around all lines from the θ to the ψ. The line with θ must be sanctioned by the rule 'Assume'.

Continued from Previous Page

Name	**Given**	**May Add**
$\leftrightarrow$-*Intro*	$\theta \rightarrow \psi$, $\psi \rightarrow \theta$	$\theta \leftrightarrow \psi$
$\leftrightarrow$-*Elim*	$\theta \leftrightarrow \psi$, ψ	θ
$\leftrightarrow$-*Elim*	$\theta \leftrightarrow \psi$, θ	ψ

Table 6.2: Basic Rules of SD

6.2.2 Boxes as an Accounting Device for Assumptions

Let's look again at the first example:

1.	$B \wedge C$	*Assume*
2.	B	$\wedge$-*Elim*, 1
3.	C	$\wedge$-*Elim*, 1
4.	$C \wedge B$	$\wedge$-*Intro*, 2,3
5.	$(B \wedge C) \rightarrow (C \wedge B)$	$\rightarrow$-*Intro*, 1–4

You will notice a box around the sentences of the first four steps. That box is the product of two rules: *Assume* and $\rightarrow$-*intro*. The *Assume* rule allows you to write any sentence you like, under one condition. You must draw a vertical line to the left of the sentence you assume, and that line must be extended to the left of all sentences of each step below until you the $\rightarrow$-*intro* rule is applied. When $\rightarrow$-*intro* is applied three more lines must be drawn, forming a box around all the sentences from the assumption to the step just before the $\rightarrow$-*intro*.

Every use of the *Assume* rule starts a box, and every use of $\rightarrow$-*intro* closes a box. The vertical line to the left represents an assumption. Every sentence that is derived on that line represents a conclusion reached using that assumption. Closing the box with $\rightarrow$-*intro* represents a *discharge* of that assumption. In other words that assumption is no longer being made. Neither the assumption nor the lines derived from that assumption may be used in the rest of the derivation. A closed the box is a visual reminder that those lines cannot be used any longer.

To understand better how assumptions work in derivations we state their use in more general terms. At any step you may assume any sentence you like, so long as you write a vertical line to the left of it: $\mid \theta$. Assumptions are discharged using the $\rightarrow$-*Intro* rule. To apply $\rightarrow$-*Intro* you need a vertical line from an unboxed assumption $\mid \theta$ that extends down to an unboxed $\mid \psi$. After you've added $\theta \rightarrow \psi$ to your derivation, you *must* box off the part of the derivation that begins with $\mid \theta$ and ends with $\mid \psi$. That is, if you have the following:

1.	θ	*Assume*
2.		
3.	$\vdots$	
4.		
5.	ψ	

you may then use the rule $\rightarrow$-*Intro* to get:

1.	θ	*Assume*
2.		
3.	$\vdots$	
4.		
5.	ψ	
6.	$\theta \rightarrow \psi$	$\rightarrow$-*Intro*, 1–5

You may even assume more than one sentence at a time, in which case you may have several vertical lines to the left. Just remember that these assumptions will typically need to be discharged.[4] Vertical assumption lines are like your credit card balance. It can be convenient to run a high balance for a time, as long as you can pay it down eventually.

6.2.3 Writing Derivations

Let's look at a few derivations in line-by-line detail. Consider the derivation of $C \wedge B$ from $B \wedge C$:

3.

1.	$B \wedge C$	*Assume*

Note that $\wedge$-*Elim* allows us to write down one of the conjuncts of a conjunction we already have. The conjuncts of $B \wedge C$ are B and C. Accordingly, we may write down B and C on new lines. We continue:

4.

1.	$B \wedge C$	*Assume*
2.	B	$\wedge$-*Elim*, 1
3.	C	$\wedge$-*Elim*, 1

Finally, using $\wedge$-*Intro* we put the sentences from lines 2 and 3 together in a conjunction:

[4] Leaving an assumption undischarged is tantamount to treating it as a premise in an argument.

5.

1.	$B \wedge C$	*Assume*
2.	B	$\wedge$-*Elim*, 1
3.	C	$\wedge$-*Elim*, 1
4.	$C \wedge B$	$\wedge$-*Intro*, 2,3

This is a four-line derivation of $C \wedge B$ from $B \wedge C$. Notice that steps 2 and 3 could have been done in opposite order. In many cases the order of sentences in a derivation doesn't matter, but in other cases it is crucial. Learning the difference is an important skill.

Just as we used the double turnstile to represent when one sentence (or a set of sentences) entailed another, we use what's called the *single turnstile*, '$\vdash$', to represent when one sentence is derivable from another, or from another (finite) set of sentences. The four-line derivation shows that $B \wedge C \vdash C \wedge B$. It's also worth noting that each of the two partially completed pieces of the derivation just given are derivations themselves. The first (3) is a one-line derivation that shows that $B \wedge C \vdash B \wedge C$, while the second (4) is a three-line derivation that shows that $B \wedge C \vdash C$.

We can now "close off" the initial assumption by drawing a box around that part of the proof and writing a conditional on the next line:

6.

1.	$B \wedge C$	*Assume*
2.	B	$\wedge$-*Elim*, 1
3.	C	$\wedge$-*Elim*, 1
4.	$C \wedge B$	$\wedge$-*Intro*, 2,3
5.	$(B \wedge C) \rightarrow (C \wedge B)$	$\rightarrow$-*Intro*, 1–4

So we have derived $(B \wedge C) \rightarrow (C \wedge B)$ without any assumptions remaining. When we have a derivation from no assumptions (or, rather, with all assumptions boxed), we represent that with a single turnstile with no formulas on the left. This derivation shows that $\vdash (B \wedge C) \rightarrow (C \wedge B)$.[5]

Next, as an example of how assumption lines stack up in a derivation, consider the following derivation of D from $A \wedge B$ and $B \rightarrow (C \wedge D)$.

[5] If $\vdash \phi$, it's often said that ϕ is a theorem of the derivation system. "Theorem" is used ambiguously between the derivation of an SL(or QL) sentence from the empty set of assumptions and things we prove in MathEnglish. We follow that standard practice.

7.

1.	$A \wedge B$	*Assume*
2.	$\quad B \rightarrow (C \wedge D)$	*Assume*
3.	$\quad B$	$\wedge$-*Elim*, 1
4.	$\quad C \wedge D$	$\rightarrow$-*Elim*, 2,3
5.	$\quad D$	$\wedge$-*Elim*, 4

As is shown, any time a new assumption line is added it must be placed to the right of the previous (unboxed) assumption lines.

We can only discharge one assumption at a time: the most recent unboxed assumption. So we get

8.

1.	$A \wedge B$	*Assume*
2	$\quad B \rightarrow (C \wedge D)$	*Assume*
3	$\quad B$	$\wedge$-*Elim*, 1
4	$\quad C \wedge D$	$\rightarrow$-*Elim*, 2,3
5	$\quad D$	$\wedge$-*Elim*, 4
6.	$(B \rightarrow (C \wedge D)) \rightarrow D$	$\rightarrow$-*Intro*, 2–5

Now that the second assumption is boxed, we are free to discharge the first:

9.

1.	$A \wedge B$	*Assume*
2.	$\quad B \rightarrow (C \wedge D)$	*Assume*
3.	$\quad B$	$\wedge$-*Elim*, 1
4.	$\quad C \wedge D$	$\rightarrow$-*Elim*, 2,3
5.	$\quad D$	$\wedge$-*Elim*, 4
6.	$(B \rightarrow (C \wedge D)) \rightarrow D$	$\rightarrow$-*Intro*, 2–5
7.	$(A \wedge B) \rightarrow ((B \rightarrow (C \wedge D)) \rightarrow D)$	$\rightarrow$-*Intro*, 1–6

We have derived $(A \wedge B) \rightarrow ((B \rightarrow (C \wedge D)) \rightarrow D)$ without any outstanding assumptions. So we have $\vdash (A \wedge B) \rightarrow ((B \rightarrow (C \wedge D)) \rightarrow D)$.

It's worth reiterating that once sentences have been boxed they can't be used for the rest of the derivation. These sentences are no longer 'Given', to use the term in the Basic Rules chart. If their use were allowed then SD would not be sound. That is, there would be some sentences ϕ and $\psi_1, \ldots, \psi_n$ such that $\psi_1, \ldots, \psi_n \not\models \phi$ but $\psi_1, \ldots, \psi_n \vdash \phi$. So $\vdash$ and $\models$ wouldn't agree with each other—an unfortunate result! To see the problem, consider:

10.

1.	$A \to B$	*Assume*
2.	$\boxed{A}$	*Assume*
3.	$\boxed{B}$	$\to$-*Elim*, 1,2
4.	$A \to B$	$\to$-*Intro*, 2–3
5.	B	$\to$-*Elim*, 2,4

Line 5 cites line 2 to use A to get B from line 4 using $\to$-*Intro*. But clearly $A \to B \models B$ does not hold. Consider a model $\mathfrak{m}$ such that $\mathfrak{m}(A) = \mathtt{F}$ and $\mathfrak{m}(B) = \mathtt{F}$.

6.2.4 The Recursive Definition of a Derivation

We have not yet given a precise definition of 'derivation'. But before we do so some prerequisite definitions are needed.

Definition 6.1. A *formal derivation rule* is a sequence of sentence schemas divided into two parts, the first part being the *given schemas* and the second part being the *may-add schema*.

Table 6.2 (on page 125) lists rules by putting the "Given" schemas in the left column, and the "May Add" schema in the right. We spell out these schemas so the next definition is easier to state.

Definition 6.2. Let there be unboxed lines $m_1, \dots, m_j$ with sentences $\psi_1, \dots, \psi_j$, respectively. A rule R, applied to lines $m_1, \dots, m_j$ *sanctions* writing the sentence ϕ iff there's some substitution of SL sentences that, for the given schemas of R, results in $\psi_1, \dots, \psi_j$ and, for the may-add schema, results in ϕ.

As an example, consider again derivation 9, on the previous page. The rule $\to$-*Elim* was applied to line 2, which had sentence $B \to (C \wedge D)$, and line 3, which had sentence B, to get line 4, which had sentence $C \wedge D$. Using definition 6.2 we can show that this move is sanctioned by $\to$-*Elim* by noting, from table 6.2 (on page 125), that $\to$-*Elim* has two given schemas, $\theta \to \psi$ and θ, and the may-add scheme ψ. Substituting $\theta = B$ and $\psi = C \wedge D$ in the given schemas gets us lines 2 and 3, while making this same substitution in the may-add schema gets us line 4.

We now give a precise definition of a derivation. The explicit definition is especially useful for proving soundness in the next chapter. Although the actual definition is complicated, the basic idea is straightforward: A single SL sentence that's an assumption is a derivation (e.g., derivation 3), and any finite sequence of SL sentences every one of which is either an assumption or sanctioned by some rule of SD is a derivation. This idea must be revised to handle the rules $\to$-*Intro* and *Assume*, neither of which quite work like the other rules. To understand these rules we must first define what an *open assumption* is. An assumption is open iff it appears on an assumption line and is not in a box.

Definition 6.3. The following recursive clauses fix which finite sequences of derivation lines are *derivations* in SD:

Base Clause: For any SL sentence ϕ, the following single derivation line (i.e., sequence of derivation lines of length 1) is a derivation:

$$\begin{array}{lll} 1. & \vert\phi & \textit{Assume} \end{array}$$

Generating Clause:

Case 1: If you have some n-line derivation with k assumptions still open at line n, and in which the sentence ψ on n is sanctioned by rule R when applied to a subset of lines $j_1, \dots, j_k$,

$$\begin{array}{lll} 1. & & \\ & \vdots & \\ n. & \psi & R,\ j_1, \dots, j_k \end{array}$$

then the sequence of derivation lines you get by adding a new line $n+1$ with k open assumptions and the sentence θ is a derivation,

$$\begin{array}{lll} 1. & & \\ & \vdots & \\ n. & \psi & R,\ j_1, \dots, j_k \\ n+1. & \theta & R',\ h_1, \dots, h_l \end{array}$$

so long as θ is sanctioned by some rule R' (other than $\rightarrow$-*Intro*) when applied to a subset of previous lines $h_1, \dots, h_l$ already in the derivation.

Case 2: If you have some n-line derivation with k assumptions still open at line n, and in which the sentence ψ on n is sanctioned by rule R when applied to a subset of lines $j_1, \dots, j_k$,

$$\begin{array}{lll} 1. & & \\ & \vdots & \\ n. & \vert\psi & R,\ j_1, \dots, j_k \end{array}$$

then, for any sentence θ, the sequence of derivation lines you get by adding a new line $n+1$ with $k+1$ open assumptions and θ sanctioned by *Assume* is a derivation,

$$
\begin{array}{lll}
1. & & \\
 & \vdots & \\
n. & \psi & R,\ j_1, \ldots, j_k \\
n+1. & \quad \theta & \textit{Assume}
\end{array}
$$

Case 3: If you have some n-line derivation with k assumptions still open at line n, and in which the sentence ψ on n is sanctioned by rule R when applied to a subset of lines $j_1, \ldots, j_k$, and in which the kth assumption was opened on line m,

$$
\begin{array}{lll}
1. & & \\
 & \vdots & \\
m. & \phi & \textit{Assume} \\
 & \quad \vdots & \\
n. & \psi & R,\ j_1, \ldots, j_k
\end{array}
$$

then the sequence of derivation lines you get by adding a new line $n+1$ with $k-1$ open assumptions and the sentence $\phi \rightarrow \psi$ is a derivation,

$$
\begin{array}{lll}
1. & & \\
 & \vdots & \\
m. & \boxed{\begin{array}{l}\phi \\ \\ \quad \vdots \\ \\ \end{array}} & \textit{Assume}
\end{array}
$$

$$\begin{array}{lll} n. & \boxed{\psi} & R,\, j_1, \ldots, j_k \\ n+1. & \phi \rightarrow \psi & \rightarrow\textit{-Intro},\, m\text{–}n \end{array}$$

so long as you close the assumption opened on line m by drawing a box around lines m–n and write down $\rightarrow$-*Intro*, m–n as the rule which sanctions line $n+1$.

Closure Clause: Nothing else is a derivation of SL.

Note that in the generating clauses we specify that we start with a derivation with k open assumptions on the last line. In cases 1 and 2 the schematic drawings given don't explicitly depict the k vertical lines that should be running between the line numbers and the sentences, but the drawings are not intended to suggest that you can write derivations without the running vertical lines which track open assumptions. The schematic drawing in case 3 similarly only depicts the last vertical assumption line (the one for the assumption opened on line m), but that's not to suggest the others aren't there, or that they can be left off.

6.2.5 Restrictions on Applying Rules

It's important to be clear on precisely when a rule sanctions writing down a sentence. The restriction, which we quietly followed in previous examples, is that a rule can be applied only if the connectives mentioned in the rule are the main connectives of the sentences to which that rule is being applied. That is, a rule can only be applied to whole sentences on a line—it can't be applied to proper subsentences of a line.

For example, we can only apply $\rightarrow$-*Elim* on two lines of a derivation if the sentence on one of the lines is a conditional $\phi \rightarrow \theta$ and the sentence on the other line is ϕ. If $\phi \rightarrow \theta$ is the sentence on one line and ϕ is merely contained as a subsentence in the sentence on the other (say the sentence has the form $\phi \vee \psi$ or $\phi \wedge \psi$), then we cannot apply $\rightarrow$-*Elim* to those lines. Likewise, if a line has a sentence ϕ and the conditional $\phi \rightarrow \theta$ is merely contained as a subsentence in the sentence on another line (say the sentence has the form $(\phi \rightarrow \theta) \vee \psi$), then we cannot apply $\rightarrow$-*Elim* to those lines.

For a more concrete example, consider derivation 9, on page 129. Even though B appears on line 1 (as the conjunct of $A \wedge B$), we cannot apply $\rightarrow$-*Elim* to lines 1 and 2 to get $C \wedge D$. The fact that we can easily get B on its own line through $\wedge$-*Elim* doesn't matter. The rule $\rightarrow$-*Elim* will only sanction writing $C \wedge D$ on a line, given $B \rightarrow (C \wedge D)$ on line 2, if we have another line with the LHS of the conditional by itself. In just the same way we cannot apply $\wedge$-*Elim* to line 2 of derivation 9 to get C or D, since the conjunction in line 2 is not the main connective. Instead, it's the main connective of the RHS subsentence of $B \rightarrow (C \wedge D)$. Again it doesn't matter that we can get the conjunct by itself using $\rightarrow$-*Elim* (after using $\wedge$-*Elim* on line 1), the rule $\wedge$-*Elim* cannot be applied to a line unless the sentence on that line is a conjunction.

6.2.6 Some Strategies

There are common strategies that can be used to make progress in derivations. For each logical connective there are two types of strategies: those for what to do if you already have sentences with that as their main connective, and those for what to do if you want to get a sentence with that as its main connective. We call the first top-down strategies and the second bottom-up strategies. In many cases, doing a derivation is like planning a plane trip. The top-down method is like figuring out the nearest convenient airport from your current location, and the bottom-up method is like figuring out which airport is convenient for getting to your destination. In derivations, sometimes you have to go through intermediate sentences (as with intermediate cities in travel).

We begin with some basic top-down and bottom-up strategies for each connective, adding more later in section 6.3.3 when we introduce shortcut rules.

Conjunction We start with the basic top-down and bottom-up strategies for conjunction. They are straightforward.

$\wedge$**Top-down:** If you have a sentence of the form $\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n$, then break it apart using $\wedge$-*Elim* to get each of the conjuncts ϕ_1 through ψ_n, each on a new line.

$\wedge$**Bottom-up:** If you want to get a sentence of the form $\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n$, then derive each of ϕ_1 through ϕ_n individually and use $\wedge$-*Intro* to derive it from them.

Both strategies are exemplified in example derivation 5, on page 128. There we wanted to derive the sentence $C \wedge B$, so in line with the bottom-up strategy for $\wedge$ we first derived both C and B and then used $\wedge$-*Intro* to derive $C \wedge B$. In line with the top-down strategy, we took our assumption $B \wedge C$ and broke it apart using $\wedge$-*Elim* (which got us the sentences, C and B, we were looking to derive).

Conditionals The basic strategies for conditionals are also straightforward and have already been exemplified.

$\rightarrow$ **Top-down:** If you have a sentence of the form $\phi \rightarrow \psi$, then first derive the LHS ϕ and then break it apart using $\rightarrow$-*Elim* to get the RHS ψ on a new line.

$\rightarrow$ **Bottom-up:** If you want to get a sentence of the form $\phi \rightarrow \psi$, then assume the LHS ϕ, derive the RHS ψ, and then use $\rightarrow$-*Intro* to write the conditional on the next line.

In derivation 7, on page 129, we wanted to derive D. We saw that we had a conditional, $B \rightarrow (C \wedge D)$. In line with the top-down strategy, we derived its LHS B (in the process using the top-down strategy for $\wedge$), then used $\rightarrow$-*Elim* to get the RHS $(C \wedge D)$. We wanted $(C \wedge D)$, of course, because from it we could use $\wedge$-*Elim* to get D. In derivation 9, on page 129, we wanted to derive $(A \wedge B) \rightarrow ((B \rightarrow (C \wedge D)) \rightarrow D)$. In line with the bottom-up strategy, we assumed the LHS $(A \wedge B)$, derived the RHS

$((B \to (C \wedge D)) \to D)$, and then used $\to$-*Intro* to write the conditional on the next line.

Biconditionals The basic strategies for biconditionals are similar to those for conditionals, as one might expect.

$\leftrightarrow$ **Top-down:** If you have a sentence of the form $\phi \leftrightarrow \psi$, then either

1. first derive the LHS ϕ and then break it apart using $\leftrightarrow$-*Elim* to get the RHS ψ on a new line,
2. first derive the RHS ψ and then break it apart using $\leftrightarrow$-*Elim* to get the LHS ϕ on a new line,
3. or do both.

$\leftrightarrow$ **Bottom-up:** If you want to get a sentence of the form $\phi \leftrightarrow \psi$, then first derive both $\phi \to \psi$ and $\psi \to \phi$ and then use $\leftrightarrow$-*Intro* to write the biconditional on the next line.

Negations In the case of negations, we don't have a basic top-down strategy, only a basic bottom-up. Later in this chapter we develop shortcut rules which allow us to provide top-down strategies for negation.

$\sim$ **Bottom-up:** If you want to get a sentence of the form $\sim\phi$, then first assume ϕ, derive a contradiction $\psi \wedge \sim\psi$, and then in two separate steps use $\to$-*Intro* and $\sim$-*Intro* to write the negation on the next line.

For example, say we want to derive the sentence $\sim((A \to \sim B) \wedge (A \wedge B))$. In line with the basic bottom-up strategy for $\sim$, we first assume $((A \to \sim B) \wedge (A \wedge B))$ and try to derive a contradiction:

11.

1.	$((A \to \sim B) \wedge (A \wedge B))$	*Assume*
	$\vdots$	
n.	$\psi \wedge \sim\psi$	

It should be clear that we can derive $B \wedge \sim B$, so that is our goal:

12.

1.	$((A \to \sim B) \wedge (A \wedge B))$	*Assume*
	$\vdots$	
n.	$B \wedge \sim B$	

The bottom-up strategy for $\wedge$ says to get this we should derive both B and $\sim B$.

13.

1.	$((A \to \sim B) \wedge (A \wedge B))$	*Assume*
	$\vdots$	
$n-2$.	B	
$n-1$.	$\sim B$	
n.	$B \wedge \sim B$	$\wedge$-*Intro*, n−1,n−2

The top-down strategy for $\wedge$ is our only option at this point, so we break line 1 apart using $\wedge$-*Elim*.

14.

1.	$((A \to \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(A \to \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
	$\vdots$	
$n-2$.	B	
$n-1$.	$\sim B$	
n.	$B \wedge \sim B$	$\wedge$-*Intro*, n−1,n−2

Now we continue to work top-down, using $\wedge$-*Elim* to break apart line 3. Note that in this step we've partway joined up the top and bottom of the proof, since breaking apart line 3 gets us what we were calling line n−2.

15.

1.	$((A \to \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(A \to \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	A	$\wedge$-*Elim*, 3
5.	B	$\wedge$-*Elim*, 3
	$\vdots$	
$n-1$.	$\sim B$	
n.	$B \wedge \sim B$	$\wedge$-*Intro*, n–1,n–2

Next we see that we can work bottom-down from lines 2 and 4, breaking the conditional on line 2 apart. In doing so we finish the proof, since the result of doing this is $\sim B$, which is all that was left to get the contradiction.

16.

1.	$((A \to \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(A \to \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	A	$\wedge$-*Elim*, 3
5.	B	$\wedge$-*Elim*, 3
6.	$\sim B$	$\to$-*Elim*, 2,4
7.	$B \wedge \sim B$	$\wedge$-*Intro*, 5,6

Of course, we haven't yet derived $\sim((A \to \sim B) \wedge (A \wedge B))$, but we can now do so by discharging the assumption through $\to$-*Intro* and then applying $\sim$-*Intro*.

17.

1.	$((A \to \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(A \to \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	A	$\wedge$-*Elim*, 3
5.	B	$\wedge$-*Elim*, 3
6.	$\sim B$	$\to$-*Elim*, 2,4
7.	$B \wedge \sim B$	$\wedge$-*Intro*, 5,6
8.	$((A \to \sim B) \wedge (A \wedge B)) \to (B \wedge \sim B)$	$\to$-Intro, 1–7
9.	$\sim((A \to \sim B) \wedge (A \wedge B))$	$\sim$-*Intro*, 8

Disjunctions Our last pair of strategies is for disjunctions. As with conjunctions, we give the strategies for the case where there are only two disjuncts. Generalizing the strategies for disjunctions with more than two disjuncts is left to the reader.

$\vee$ **Top-down:** If you have a sentence of the form $\phi \vee \psi$ and you want to derive a sentence θ, first derive the conditionals $\phi \rightarrow \theta$ and $\psi \rightarrow \theta$, and then use $\vee$-*Elim* to write down θ on the next line. The order in which you derive the intermediate conditionals doesn't matter.

$\vee$ **Bottom-up:** If you want a sentence of the form $\phi \vee \psi$, then first derive either ϕ or derive ψ, and then use $\vee$-*Intro* to write it down.

The basic bottom-up strategy isn't always the right tool for deriving disjunctions, because usually you can't derive one of the disjuncts. This is an important point: it might be that a disjunction is derivable even if neither disjunct is. It's important that we can derive disjunctions without first deriving one or the other disjunct, since $B \vee \sim B$ is a logical truth. We would like to be able to derive it, though neither B nor $\sim B$ is a logical truth.

Later on we get more useful bottom-up strategies for disjunctions. For now we focus on the top-down strategy.

As an example, say we want to derive $\sim((\sim A \vee \sim B) \wedge (A \wedge B))$. The whole sentence itself is a negation, so we start just as in the last example. So we need some contradiction to aim for. We try to derive $B \wedge \sim B$.

18.

1.	$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
	$\vdots$	
n.	$B \wedge \sim B$	

As in the last example the bottom-up strategy for $\wedge$ has us try to derive both B and $\sim B$.

19.

1.	$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
	$\vdots$	
$n-2$.	B	
$n-1$.	$\sim B$	
n.	$B \wedge \sim B$	$\wedge$-*Intro*, $n-2$,$n-1$

And, the top-down strategy leads us to break apart the conjunction on line 1, which leads to another conjunction to break apart as well. This gets us one of the conjuncts of line n in the process.

20.

1.	$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	B	$\wedge$-*Elim*, 3
	$\vdots$	
$n-1$.	$\sim B$	$\sim$-*Intro*, $n-2$
n.	$B \wedge \sim B$	$\wedge$-*Intro*, $n-2$,$n-1$

Now we only need to get $\sim B$. To do this, we follow the bottom-up strategy for negation. We assume B and try to get a contradiction.

21.

1.	$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	B	$\wedge$-*Elim*, 3
5.	B	*Assume*
	$\vdots$	
$n-3$.	$\psi \wedge \sim\psi$	
$n-2$.	$B \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 5–$n-3$
$n-1$.	$\sim B$	$\sim$-*Intro*, $n-2$
n.	$B \wedge \sim B$	$\wedge$-*Intro*, $n-2$,$n-1$

Note that although it may look wrong to have an assumption line for B when we have already derived it, there's nothing wrong with this derivation. You can always assume whatever you want, even if you already have it. Now we have two questions to think about: what contradiction could we get on line $n-3$? And, how do we get it? The second question is straightforward. We want to get a contradiction, and at this point it's going to have to come from the disjunction on line 2. So we follow the top-down strategy for $\vee$ which tells us how to get a sentence from a disjunction.

22.

1.	$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	B	$\wedge$-*Elim*, 3
5.	B	*Assume*
	$\vdots$	
$n-5$.	$\sim B \rightarrow (\psi \wedge \sim\psi)$	
$n-4$.	$\sim A \rightarrow (\psi \wedge \sim\psi)$	
$n-3$.	$\psi \wedge \sim\psi$	$\vee$-*Elim*, 2,$n-5$,$n-4$
$n-2$.	$B \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 5–$n-3$
$n-1$.	$\sim B$	$\sim$-*Intro*, $n-2$
n.	$B \wedge \sim B$	$\wedge$-*Intro*, $n-2$,$n-1$

To get the conditionals on lines $n-5$ and $n-4$, we have to use $\rightarrow$-*Into*. So the setup is:

23.

1.	$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	B	$\wedge$-*Elim*, 3
5.	B	*Assume*
6.	$\sim B$	*Assume*
	$\vdots$	
m.	$(\psi \wedge \sim\psi)$	
$m+1$.	$\sim B \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 6–m
$m+2$.	$\sim A$	*Assume*
	$\vdots$	
$n-5$.	$(\psi \wedge \sim\psi)$	

$n-4.$	$\sim A \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, $m+2$–$n-5$
$n-3.$	$\psi \wedge \sim\psi$	$\vee$-*Elim*, 2,$m+1$,$n-4$
$n-2.$	$B \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 5–$n-3$
$n-1.$	$\sim B$	$\sim$-*Intro*, $n-2$
$n.$	$B \wedge \sim B$	$\wedge$-*Intro*, $n-2$,$n-1$

Now we can decide what contradiction $(\psi \wedge \sim\psi)$ to aim for. If we choose $(B \wedge \sim B)$ again, then the first conditional is easy. We are able to get it by using $\wedge$-*Intro* on lines 5 and 6.

24.

1.	$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
2.	$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	B	$\wedge$-*Elim*, 3
5.	B	*Assume*
6.	$\sim B$	*Assume*
7.	$(B \wedge \sim B)$	$\wedge$-*Intro*, 5,6
8.	$\sim B \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 6–7
9.	$\sim A$	*Assume*
	$\vdots$	
$n-5.$	$(B \wedge \sim B)$	
$n-4.$	$\sim A \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 9–$n-5$
$n-3.$	$B \wedge \sim B$	$\vee$-*Elim*, 2,8,$n-4$
$n-2.$	$B \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 5–$n-3$
$n-1.$	$\sim B$	$\sim$-*Intro*, $n-2$
$n.$	$B \wedge \sim B$	$\wedge$-*Intro*, $n-2$,$n-1$

This leaves us with just the second conditional, deriving $(B \wedge \sim B)$ from $\sim A$. At first glance this may seem impossible. We were able to derive $(B \wedge \sim B)$ from $\sim B$ because, given that we already had B, assuming $\sim B$ allowed us to use $\wedge$-*Intro*. But $\sim A$ obviously isn't sufficient to allow us to use $\wedge$-*Intro*, and there's no clear way to get what we need for $\wedge$-*Intro*, $\sim B$, from $\sim A$. So we won't be able to get $(B \wedge \sim B)$ by using $\wedge$-*Intro*.

Now so far we only have one strategy for getting a conjunction and that's to derive the conjuncts and use $\wedge$-*Intro*. But there's another strategy, not tied to any

particular connective, which we can use here. This strategy is to use $\sim$-*Elim*. We assume $\sim(B \wedge \sim B)$, derive a contradiction (any will do), and then use $\rightarrow$-*Intro* to get what we need. It should be reasonably clear that this strategy will work, since we obviously can derive the contradiction $(A \wedge \sim A)$.

25.

1.		$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
2.		$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.		$(A \wedge B)$	$\wedge$-*Elim*, 1
4.		B	$\wedge$-*Elim*, 3
5.		B	*Assume*
6.		$\sim B$	*Assume*
7.		$(B \wedge \sim B)$	$\wedge$-*Intro*, 5,6
8.		$\sim B \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 6–7
9.		$\sim A$	*Assume*
10.		$\sim(B \wedge \sim B)$	*Assume*
11.		A	$\wedge$-*Elim*, 3
12.		$A \wedge \sim A$	$\wedge$-*Intro*, 9,11
13.		$\sim(B \wedge \sim B) \rightarrow (A \wedge \sim A)$	$\rightarrow$-*Intro*, 10–12
14.		$(B \wedge \sim B)$	$\sim$-*Elim*, 13
15.		$\sim A \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 9–14
16.		$B \wedge \sim B$	$\vee$-*Elim*, 2,8,15
17.		$B \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 5–16
18.		$\sim B$	$\sim$-*Intro*, 17
19.		$B \wedge \sim B$	$\wedge$-*Intro*, 4,18

Note that we didn't actually use the assumption $\sim(B \wedge \sim B)$ from line 10 in deriving the contradiction we end with on line 12, $A \wedge \sim A$. There is nothing wrong with this, since $\rightarrow$-*Intro* doesn't require that the sentence θ with which you started is actually used in deriving the sentence ψ with which you finished.

Now that we've derived a contradiction from $((\sim A \vee \sim B) \wedge (A \wedge B))$ we can finish the derivation by discharging the assumption with $\rightarrow$-*Intro* and then use $\sim$-*Intro*.

26.

1.		$((\sim A \vee \sim B) \wedge (A \wedge B))$	*Assume*
2.		$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.		$(A \wedge B)$	$\wedge$-*Elim*, 1
4.		B	$\wedge$-*Elim*, 3

5.	B	*Assume*
6.	$\sim B$	*Assume*
7.	$(B \wedge \sim B)$	$\wedge$-*Intro*, 5,6
8.	$\sim B \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 6–7
9.	$\sim A$	*Assume*
10.	$\sim(B \wedge \sim B)$	*Assume*
11.	A	$\wedge$-*Elim*, 3
12.	$A \wedge \sim A$	$\wedge$-*Intro*, 9,11
13.	$\sim(B \wedge \sim B) \rightarrow (A \wedge \sim A)$	$\rightarrow$-*Intro*, 10–12
14.	$(B \wedge \sim B)$	$\sim$-*Elim*, 13
15.	$\sim A \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 9–14
16.	$B \wedge \sim B$	$\vee$-*Elim*, 2,8,15
17.	$B \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 5–16
18.	$\sim B$	$\sim$-*Intro*, 17
19.	$B \wedge \sim B$	$\wedge$-*Intro*, 4,18
20.	$((\sim A \vee \sim B) \wedge (A \wedge B)) \rightarrow (B \wedge \sim B)$	$\rightarrow$-*Intro*, 1–19
21.	$\sim((\sim A \vee \sim B) \wedge (A \wedge B))$	$\sim$-*Intro*, 20

Proof by Contradiction A general strategy involving negation was used in derivation 26, on the previous page. This strategy formalizes an informal proof method often called proof by contradiction. In proof by contradiction, one proves that some sentence ϕ is true by assuming it's false and showing that a contradiction follows. The corresponding strategy in SD is:

Proof by Contradiction: If you want to get some sentence ϕ, then first derive $\sim\phi \rightarrow (\psi \wedge \sim\psi)$ and then use $\sim$-*Elim* to get ϕ from this conditional.

This strategy has the virtue that if you can derive ϕ (without any assumptions), then you are able to derive a contradiction $(\psi \wedge \sim\psi)$ from $\sim\phi$ as an assumption. In other words, the strategy will always work. But despite this, there are usually much faster and better ways to derive a sentence. For example, write a derivation of $C \wedge B$ from $B \wedge C$ using proof by contradiction and then compare it to derivation 5, on page 128. There are some cases where proof by contradiction is the *only* strategy that will work. So there are two cases where it is a good idea to use the strategy: cases where all other available strategies haven't worked (or where there aren't any other available strategies), and cases where you can see a straightforward way to use it.

Top-down and Bottom-up Finally, the reader should reflect on the general method we have been using. In slogan form, the method goes: work top-down *and* bottom-up.

When writing derivation it's important to work not only from the assumptions, seeing how you can move down from them to the conclusion using the rules, but also to work up from the sentence you're trying to derive, looking for the different ways you can get there. Hence the grouping of strategies into top-down and bottom-up. The top-down strategies help guide what moves you can make as you work from the assumptions to the conclusion, while the bottom-up strategies help see what different paths there are to get to that conclusion.

As an aside, the top-down and bottom-up method is something useful outside of writing formal derivations. While most arguments, including those found in philosophy, mathematical proofs, and formal derivations, are presented as a series of steps from premises to conclusion, they are seldom devised that way. Usually a mathematician starts with a conjecture and tries to work "up" from it to results that have already been proven. Rarely does one from the start know what premises are needed to prove a conjecture. Something similar goes for philosophy and other disciplines that rely on argument. So, the reader can think of the top-down, bottom-up method used in derivations as a reflection of how informal arguments and proofs are constructed in philosophy, mathematics, and other disciplines.

6.3 Shortcut Rules for SD

6.3.1 Standard Shortcut Rules

From derivation 26, on page 142, we can see that derivations in SD of even simple looking sentences can be long and involve roundabout strategies. This is the main reason why we want to introduce shortcut rules. Shortcut rules can be thought of as ways of cutting out parts of derivations that we find ourselves doing repeatedly. The basic rules of SD plus the shortcut rules (both those in table 6.39 on the next page and table 6.40 on page 146) make up the derivation system which we call SD^+.

For example, consider lines 10–14 of proof 26, on page 142. Here we have two sentences, A and $\sim A$, and we used them to derive the sentence $B \wedge \sim B$. We can rewrite the relevant parts of the proof here, putting them in a less idiosyncratic order:

27.

1.	A	
2.	$\sim A$	
3.	$\boxed{\sim(B \wedge \sim B)}$	*Assume*
4.	$\boxed{A \wedge \sim A}$	$\wedge$*-Intro, 1,2*
5.	$\sim(B \wedge \sim B) \rightarrow (A \wedge \sim A)$	$\rightarrow$*-Intro*, 3–4
6.	$B \wedge \sim B$	$\sim$*-Elim*

It should be clear from looking at this derivation that we can replace $B \wedge \sim B$ with any sentence ψ and the string of sentences that results from this replacement will also

be a derivation. It should also be clear that we can replace A and $\sim A$ with any pair ϕ and $\sim\phi$ and the resulting string of sentences is a derivation. That is,

28.

1.	ϕ	
2.	$\sim\phi$	
3.	$\boxed{\sim\psi}$	*Assume*
4.	$\boxed{\phi \wedge \sim\phi}$	*∧-Intro, 1,2*
5.	$\sim\psi \rightarrow (\phi \wedge \sim\phi)$	*→-Intro*, 3–4
6.	ψ	*∼-Elim*

is a derivation of ψ for any sentences ϕ and ψ. More importantly, any time we're doing a derivation and we have two lines, one with a sentence ϕ and the other with its negation $\sim\phi$, we could insert a derivation of just this form to get a sentence ψ. So we can introduce a new rule which says that given two sentences ϕ and $\sim\phi$, we can add any sentence ψ. We call this new rule *Any Contradiction*, or *A.C.* for short. The key feature of this rule is that in any derivation where we use the rule, we could have gotten the same results without it. All we'd have to do is insert the appropriate instance of 28 into the derivation. (We leave it to the reader to rewrite derivation 26 on page 142 using *A.C.*)

Name	**Given**	**May Add**
M.T.	$\phi \rightarrow \theta$, $\sim\theta$	$\sim\phi$
D.S.	$\phi_1 \vee ... \vee \phi_i \vee ... \vee \phi_n$, $\sim\phi_i$	$\phi_1 \vee ... \vee \phi_{i-1} \vee \phi_{i+1} \vee ... \vee \phi_n$
	$\phi_1 \vee ... \vee \sim\phi_i \vee ... \vee \phi_n$, ϕ_i	$\phi_1 \vee ... \vee \phi_{i-1} \vee \phi_{i+1} \vee ... \vee \phi_n$
A.C.	$\phi, \sim\phi$	ψ
∼/↔-Intro	$\phi \leftrightarrow \psi$	$\sim\phi \leftrightarrow \sim\psi$
Ext. ∧-Elim	$\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n$	Conjunction of any subset of the conjuncts

Table 6.39: Standard Shortcut Rules for SD

The idea is the same for all the shortcut rules we introduce here. These come in two types, standard and exchange, and are listed in table 6.39, on this page and 6.40, on the next page. The idea is, (i) for each shortcut rule R, for given any application of R we can derive, using only the basic rules, the sentence ϕ (which R allows us to write down) from the sentences $\psi_1, ..., \psi_n$ to which we applied R. And, (ii) in general, anything we can derive using a rule, any application of which can be derived using

only basic rules, can itself be derived using only basic rules. So, (iii) anything we can derive using the basic rules of SD and any of the shortcut rules can be derived from just the basic rules alone. We have restated (ii) more precisely below as theorem 6.5 (on the next page), (i) as theorem 6.6 (on page 148), and (iii) as theorem 6.7 (on page 148). Exchange rules are short cut rules that work in both directions.

Name	**Given**	**May Add**
DeM	$\sim(\phi_1 \wedge ... \wedge \phi_n)$	$\sim\phi_1 \vee ... \vee \sim\phi_n$
	$\sim\phi_1 \vee ... \vee \sim\phi_n$	$\sim(\phi_1 \wedge ... \wedge \phi_n)$
	$\sim(\phi_1 \vee ... \vee \phi_n)$	$\sim\phi_1 \wedge ... \wedge \sim\phi_n$
	$\sim\phi_1 \wedge ... \wedge \sim\phi_n$	$\sim(\phi_1 \vee ... \vee \phi_n)$
$\sim\sim$*-Elim*	$\sim\sim\phi$	ϕ
$\sim\sim$*-Intro*	ϕ	$\sim\sim\phi$
$\rightarrow/\vee$*-Exch.*	$\phi \rightarrow \theta$	$\sim\phi \vee \theta$
	$\sim\phi \vee \theta$	$\phi \rightarrow \theta$
Contraposition	$\phi \rightarrow \theta$	$\sim\theta \rightarrow \sim\phi$
	$\sim\theta \rightarrow \sim\phi$	$\phi \rightarrow \theta$
$\sim/\rightarrow$*-Exch.*	$\sim(\phi \rightarrow \theta)$	$\phi \wedge \sim\theta$
	$\phi \wedge \sim\theta$	$\sim(\phi \rightarrow \theta)$
Distribution	$\theta \wedge (\phi_1 \vee ... \vee \phi_n)$	$(\theta \wedge \phi_1) \vee ... \vee (\theta \wedge \phi_n)$
	$(\theta \wedge \phi_1) \vee ... \vee (\theta \wedge \phi_n)$	$\theta \wedge (\phi_1 \vee ... \vee \phi_n)$
	$(\phi_1 \vee ... \vee \phi_n) \wedge \theta$	$(\phi_1 \wedge \theta) \vee ... \vee (\phi_n \wedge \theta)$
	$(\phi_1 \wedge \theta) \vee ... \vee (\phi_n \wedge \theta)$	$(\phi_1 \vee ... \vee \phi_n) \wedge \theta$
	$\theta \vee (\phi_1 \wedge ... \wedge \phi_n)$	$(\theta \vee \phi_1) \wedge ... \wedge (\theta \vee \phi_n)$
	$(\theta \vee \phi_1) \wedge ... \wedge (\theta \vee \phi_n)$	$\theta \vee (\phi_1 \wedge ... \wedge \phi_n)$
	$(\phi_1 \wedge ... \wedge \phi_n) \vee \theta$	$(\phi_1 \vee \theta) \wedge ... \wedge (\phi_n \vee \theta)$
	$(\phi_1 \vee \theta) \wedge ... \wedge (\phi_n \vee \theta)$	$(\phi_1 \wedge ... \wedge \phi_n) \vee \theta$

Table 6.40: Exchange Short-Cut Rules for SD

Theorem 6.5 (on the next page) is a general claim that doesn't require a different

proof for each new rule we introduce. The proof for it fills out the quick argument given when *A.C.* was introduced. There we argued that since any application of *A.C.* can be derived using just the basic rules, we can eliminate that application of the rule from the proof by cutting and pasting the derivation of that application into the original proof.

Definition 6.4. Every application of a rule R_1 is derivable using the rules R_2, ..., R_p and the basic rules of SD iff for all SL sentences $\phi_1, \ldots, \phi_m$ and ψ, if R_1 sanctions writing down ψ when applied to $\phi_1, \ldots, \phi_m$ on previous unboxed lines, then ψ can be derived from $\phi_1, \ldots, \phi_m$ using only rules $R_2, \ldots, R_p$ and the basic rules of SD.

Theorem 6.5. For all SL sentences $\theta_1, \ldots, \theta_n, \delta$ and rules $R_1, \ldots, R_p$, if

(1) δ can be derived from $\theta_1, \ldots, \theta_n$ using rules $R_1, \ldots, R_p$ and the basic rules of SD, and

(2) every application of a rule R_1 is derivable using the rules R_2, ..., R_p and the basic rules of SD,

then δ can be derived from $\theta_1, \ldots, \theta_n$ using only rules $R_2, \ldots, R_p$ and the basic rules of SD.

Proof: Call the original derivation of δ from $\theta_1, \ldots, \theta_n$ using rules $R_1, \ldots, R_p$ derivation D_1. Say the first use of R_1 happens in D_1 on line q. Say in that case sentence ψ was written down (on line q) and the application of the rule used previous lines $r_1, \ldots, r_m$ with sentences $\phi_1, \ldots, \phi_m$, respectively, on them. By assumption, ψ can be derived from sentences $\phi_1, \ldots, \phi_m$ using only $R_2, \ldots, R_p$ and the basic rules of SD. Call this derivation D^* and assume $\phi_1, \ldots, \phi_m$ are, respectively, on lines 1 through m in D^*. Assume there are s more lines (call these the middle lines of D^*), and finally on line $m + s + 1$ of D^* is ψ.

Now in between lines $q - 1$ and q of D_1 insert s new lines. On these lines put the appropriate sentence from the s middle line of D^*. (So, put the sentence on the first middle line of D^* on the first new line inserted into D_1, the second on the second, etc.) Write the same rules as justifications on these new lines as were on the middle lines of D^* and for each justification if t was the number of a line cited by that justification in D^*, cite line number r_t if $t \leq m$ and cite line number $(t - m) + (q - 1)$ if $t > m$. Next, on the line originally numbered q (now numbered $q + s$) erase rule R_1 as the justification and whatever line numbers t were cited and in place of that put whatever rule was used to justify the last line of D^*, citing instead lines r_t if $t \leq m$ and lines $(t - m) + (q - 1)$ if $t > m$. Finally, on all the lines of D_1 that were originally numbered q or higher (and now are numbered $q + s$ and higher), if the justification cites line t for $t < q$, do nothing. If it cites line t for $t \geq q$, replace t with $t + s$.

Note that we've now produced a derivation D_2 of δ from $\theta_1, \ldots, \theta_n$ that has one less application of R_1 than D_1 had. Now if there is an application of rule R_1 in D_2, repeat exactly this procedure for D_2. Repeating the procedure will lead to a derivation D_3

with one less application of R_1 than D_2 had. We can continue this, producing some series $D_1, D_2, \dots, D_l$ of derivations which will eventually end in a derivation D_l that has no applications of R_1. (This procedure must end, since there could only have been a finite number of applications of R_1 in D_1.) Thus, D_l is a derivation of δ from $\theta_1, \dots, \theta_n$ that uses only rules $R_2, \dots, R_p$ and the basic rules of SD. ■

Theorem 6.6. For all standard and exchange shortcut rules R (see tables 6.39 and 6.40), every application of R is derivable using the basic rules of SD.

Proof: See the discussion immediately following the proof of theorem 6.7. ■

Theorem 6.7. Shortcut Rule Elimination Theorem: For all SL sentences $\phi_1, \dots, \phi_m$ and ψ, if ψ can be derived from $\phi_1, \dots, \phi_m$ in SD$^+$ (that is, using the basic rules of SD and any of the standard and exchange shortcut rules), then ψ can be derived from $\phi_1, \dots, \phi_m$ in SD (that is, using only the basic rules).

Proof: Assume that ψ can be derived from $\phi_1, \dots, \phi_m$ using the basic rules of SD and the standard and exchange shortcut rules. Consider any of the shortcut rules, say *M.T.* Let R_1 be *M.T.* and rules R_2 through R_{25} be the other standard and exchange rules. By assumption, condition (1) in theorem 6.5 (on the previous page) holds for these sentences, while by theorem 6.6 (on the current page) condition (2) in theorem 6.5 holds for *M.T.* So, it follows from theorem 6.5 that ψ can be derived from $\phi_1, \dots, \phi_m$ using the basic rules of SD and all the standard and exchange shortcut rules besides *M.T.* By reapplying theorem 6.5 in just the same way to all the standard and exchange shortcut rules, we get that ψ can be derived from $\phi_1, \dots, \phi_m$ using only the basic rules of SD. (That is, we reapply theorem 6.5 twenty four more times, each time showing that another shortcut rule wasn't needed.) ■

Unlike theorem 6.5, for theorem 6.6 we need a separate argument for each shortcut rule (both standard and exchange). We've already done one rule, *Any Contradiction*. Our argument in this case was that any application of the rule will involve writing some sentence ψ on a new line from sentences ϕ and $\sim\phi$. But whatever sentences ψ and ϕ we pick, if we substitute them into 28, the result is a derivation of ψ from ϕ and $\sim\phi$.

Before continuing, it's important to note that, strictly speaking, 28 is *not* a derivation. This is because a derivation, as we've defined it, is a series of SL sentences. The strings of symbols on each line of 28 are not SL sentences because they contain MathEnglish variables for SL sentences. Instead, they are sentence schemas. But, as we've done in 28, nothing stops us from treating sentence schemas like the ones in 28 as SL sentences and applying rules to them. The key fact—why this is useful—is that what we get when we do this becomes a derivation whenever we substitute SL sentences in for the MathEnglish variables. For this reason we call these *derivation schemas* instead of derivations.

Returning to theorem 6.6 (on the previous page), we can handle the other rules in just the same way we handled *Any Contradiction*. For example, if we write a derivation schema of $\sim\phi$ from $\phi \to \theta$ and $\sim\theta$ using only basic rules of SD, then this is sufficient to show that any application of the rule *M.T.* (*Modus Tollens*) can be derived using only basic rules of SD.[6] (Recall def. 6.4, on page 147: Saying that an application of a rule can be derived using only basic rules is a shorthand way of saying that the sentence written down on a new line, in some application of the rule, can be derived using only basic rules from the sentences to which the rule was applied.) So, in order to complete the proof for theorem 6.6, we need to write derivation schemas for all the standard and exchange rules (for all the rules in tables 6.39 and 6.40). That is, for each rule we need to write a derivation schema that has the given schemas of the rule as premises and the may-add schema of the rule as conclusion.

Note that once we have shown that a shortcut rule can be eliminated from a proof (i.e., once we've shown that theorem 6.6, holds at least for that rule), then we can use that shortcut rule in derivation schemas for new shortcut rules we haven't yet shown can be eliminated. If we write a derivation schema for some new shortcut rule we're trying to show can be eliminated and that schema uses a previous shortcut rule, then any derivation got from the schema will contain an application of the previous rule. But we already know that these applications of the previous rule can be eliminated, so that's not a problem.

Most of the derivation schemas for the shortcut rules (both standard and exchange) are left to the reader as exercises. (See section 6.4.2.) There is one complication though. We cannot actually write a single derivation schema for *D.S.* (*Disjunctive Syllogism*), *DeM* (*DeMorgans*), or *Distribution*. This is because these rules mention arbitrarily long conjunctions and disjunctions and we can only write a derivation schema involving conjunctions and disjunctions of definite, fixed (and finite) length. A less rigorous option is to write a few derivation schemas for *D.S.*, *DeM*, and *Distribution* for the cases when the conjunctions and disjunctions are small (say 2- or 3-place) and convince ourselves that we can keep writing similar schemas no matter how large the conjunctions and disjunctions get. A more rigorous option is to write the derivation schema for the 2-place case and then use mathematical induction on the length of the conjunctions and disjunctions to show derivation schemas can be written for all lengths. In the exercises, in section 6.4.2, we only ask the reader to write the derivation schemas for these three rules for the cases where the conjunctions and disjunctions are 2-place.

Here we write the derivation schema needed for the last of the four *DeMorgans* rules in table 6.40 (on page 146), assuming that the conjunction and disjunction are only 2-place. So we need to derive $\sim(\phi \vee \theta)$ from $\sim\phi \wedge \sim\theta$. The sentence we want is

[6] For those keeping track of the use/mention distinction, here we have mentioned the MathEnglish variables 'ϕ' and 'θ' as well as the strings of symbols '$\sim\phi$', '$\phi \to \theta$', and '$\sim\theta$'. So, strictly speaking, we should have put them all in quotes. This is different how we normally use these symbols, since we're normally actually using them (as variables) instead of mentioning them (as the objects of derivation schemas).

a negation, so we use our basic bottom-up strategy for negation:

29.

1.	$\sim\phi \wedge \sim\theta$	*Assume*
2.	$(\phi \vee \theta)$	*Assume*
	⋮	
$n-2.$	$\psi \wedge \sim\psi$	
$n-1.$	$(\phi \vee \theta) \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 2–$n-2$
$n.$	$\sim(\phi \vee \theta)$	$\sim$-*Intro*, $n-1$

As always with $\sim$-*Intro*, we need some contradiction $\psi \wedge \sim\psi$. Before we had to be careful about setting up the right contradiction (recall derivation 26). But now that we have rule *A.C.*, we don't need to be so careful. So long as we can get a contradiction, we can always use *A.C.* to get whatever other contradiction we need to make the derivation work.

Returning to the proof, we need to work top-down from the disjunction on line 2 to get to a contradiction. So we use the basic strategy:

30.

1.	$\sim\phi \wedge \sim\theta$	*Assume*
2.	$(\phi \vee \theta)$	*Assume*
	⋮	
$n-4.$	$\phi \rightarrow (\psi \wedge \sim\psi)$	
$n-3.$	$\theta \rightarrow (\psi \wedge \sim\psi)$	
$n-2.$	$\psi \wedge \sim\psi$	$\vee$-*Elim*, 2,$n-4$,$n-3$
$n-1.$	$(\phi \vee \theta) \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 2–$n-2$
$n.$	$\sim(\phi \vee \theta)$	$\sim$-*Intro*, $n-1$

And from here we need to work bottom-up from the conditionals on lines $n-4$ and $n-3$.

31.

1.	$\sim\phi \wedge \sim\theta$	*Assume*
2.	$(\phi \vee \theta)$	*Assume*
3.	ϕ	*Assume*
	$\vdots$	
m.	$(\psi \wedge \sim\psi)$	
$m+1$.	$\phi \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 3-m
$m+2$.	θ	*Assume*
	$\vdots$	
$n-4$.	$(\psi \wedge \sim\psi)$	
$n-3$.	$\theta \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, $m+2$–$n-4$
$n-2$.	$\psi \wedge \sim\psi$	$\vee$-*Elim*, 2,$m+1$,$n-3$
$n-1$.	$(\phi \vee \theta) \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 2–$n-2$
n.	$\sim(\phi \vee \theta)$	$\sim$-*Intro*, $n-1$

And now we can finish the proof by working top-down, breaking apart the conjunction on line 1 and using *A.C.*

32.

1.	$\sim\phi \wedge \sim\theta$	*Assume*
2.	$(\phi \vee \theta)$	*Assume*
3.	ϕ	*Assume*
4	$\sim\phi$	$\wedge$-*Elim*, 1
5	$\sim\phi \wedge \phi$	$\wedge$-*Intro*, 3,4
6.	$(\psi \wedge \sim\psi)$	*A.C.*, 5
7.	$\phi \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 3–6
8.	θ	*Assume*
9	$\sim\theta$	$\wedge$-*Elim*, 1
10	$\sim\theta \wedge \theta$	$\wedge$-*Intro*, 8,9
11.	$(\psi \wedge \sim\psi)$	*A.C.*, 10
12.	$\theta \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 8–11
13.	$\psi \wedge \sim\psi$	$\vee$-*Elim*, 2,7,12

14.	$(\phi \vee \theta) \rightarrow (\psi \wedge \sim\psi)$	$\rightarrow$-*Intro*, 2–13
15.	$\sim(\phi \vee \theta)$	$\sim$-*Intro*, 14

And so we now have a derivation schema showing how to derive a sentence of the form $\sim(\phi \vee \theta)$ from one of the form $\sim\phi \wedge \sim\theta$. Note that we could have slightly shortened the derivation schema by doing $\rightarrow$-*Intro* on lines 3–5 instead of 3–6, using $\sim\phi \wedge \phi$ as our contradiction instead of $\phi \rightarrow (\psi \wedge \sim\psi)$. In this case would have then used *A.C.* on line 11 to get $\sim\phi \wedge \phi$. Similarly, we could have also used $\sim\theta \wedge \theta$ as our contradiction. We leave it to the reader to show what slight modifications would be needed to the derivation schema in this case.

Finally, we should note that every application of every shortcut rule (whether a standard or exchange shortcut rule) is truth-preserving. (Recall def. 8.1, on page 179.) Although you might be tempted to try to prove this as a corollary to theorem 8.2, on page 179, and theorem 6.6, on page 148, there's no easy way to get from (i) the claim that every application of every basic rule of SD is truth-preserving and (ii) the claim that every application of every rule of SD$^+$ is derivable in SD, to the further claim that every application of every rule of SD$^+$ is truth-preserving. Instead, the easiest way to prove this result more or less follows the lines of the proof for theorem 8.2 (on page 179).

Theorem 6.8. Every application of every rule of SD$^+$, including both standard and exchange shortcut rules, is truth-preserving.

Proof: Adjusting the proof of theorem 8.2 to work for the rules of SD$^+$ is left for the reader. The key is extending the truth-preservation lemma mentioned there to the rules of SD$^+$. We asked the reader to show this in exercises 2.8.8 (on page 55) and 2.8.9 (on page 55). ■

6.3.2 Exchange Shortcut Rules

Recall our restriction in section 6.2.5 on the basic rules of SD, which said that a rule only sanctions writing down a sentence if the connectives it mentions are the main connectives of sentences on the lines to which it's applied. We put this restriction in place because not every application of the basic rules is truth-preserving without it. But with the restriction in place, every application of the basic rules becomes truth-preserving (theorem 8.2, on page 179). When we introduced the shortcut rules we carried over the restriction because, again, without it there are some shortcut rules with nontruth-preserving applications. But for some of our shortcut rules, the exchange shortcut rules (table 6.40, on page 146), every application is truth-preserving even without the restriction. To be explicit, for the exchange shortcut rules we can use the following definition of sanctioning (while still using def. 6.2 on page 130 for the basic rules and the standard exchange rules):

Definition 6.9. An exchange shortcut rule R from SD$^+$, applied to a line with sentence ψ, *sanctions* writing down sentence ψ^* iff

(1) there is some substitution of SL sentences that, for the given schema of R, results in a sentence ϕ and, for the may-add schema, results in a sentence ϕ^*,

(2) ϕ is a subsentence of ψ, and

(3) ψ^* is the SL sentence you get when you replace one instance (token) of ϕ with an instance (token) of ϕ^* in ψ.

Note that right below derivation 34 (on the following page) is a detailed example of how the definition works in practice. With a moments thought it should be clear that if ϕ actually is ψ, then (recalling that all the exchange rules only have one given schema) this definition lines up with the original definition 6.2 (on page 130). This is just what we would expect.

Theorem 6.10. Every application of every exchange shortcut rule from SD$^+$ is truth-preserving, even if we extend the notion of sanctioning for them with definition 6.9.

Note that this result was not already stated in theorem 6.8 (on the previous page). Although theorem 6.8 says that every application of every rule of SD$^+$ is truth-preserving, truth-preservation is defined in terms of sanctioning (def. 8.1). And, in theorem 6.8 it was assumed that the exchange shortcut rules only sanctioned writing down a sentence if they were applied to whole sentences on lines. But now we're extending the notion of sanctioning for the exchange shortcut rules with definition 6.9. Theorem 6.10 notes that even if we do this, the exchange shortcut rules are still truth-preserving.

We use theorem 2.72 (on page 45) and the following theorem to prove theorem 6.10. This theorem will play the same rule as the truth-preservation lemma from the proof of theorem 8.2 (on page 179).

Theorem 6.11. For all exchange shortcut rules R from SD$^+$, if ϕ and ϕ^* are the sentences you get after substituting SL sentences into the given and may-add schemas of R, respectively, then ϕ and ϕ^* are truth functionally equivalent.

Proof: This theorem follows immediately from what the reader showed in exercises 2.8.9 (on page 55). ■

Proof of Thm. 6.10: Consider some arbitrary application of some exchange shortcut rule R from SD$^+$. Say that in this application R is applied to sentence ψ and permits, or sanctions, you to write down ψ^*. By definition 6.9 (on this page), (i) there is some substitution of SL sentences that, for the given schema of R, results in a sentence ϕ and, for the may-add schema, results in a sentence ϕ^*, (ii) ϕ is a subsentence of ψ, and (iii) ψ^* is the SL sentence you get when you replace one instance (token) of ϕ with an instance (token) of ϕ^* in ψ. From (i) and theorem 6.11, ϕ and ϕ^* are truth functionally equivalent. So, from theorem 2.72 (on page 45) it follows that ψ and ψ^*

are also truth functionally equivalent. From the definition 2.49 (on page 34) of truth functional equivalence it follows that $\psi \models \psi^*$. So, by definition 8.1 (on page 179), this application is truth-preserving. ■

The intuitive idea behind definition 6.9 (on the previous page) is that the exchange shortcut rules can be applied to subsentences on lines (while the basic rules and standard shortcut rules can only be applied to whole sentences on lines). Some examples will hopefully make things more clear. Recall derivation 26, on page 142, which showed that $\vdash \sim((\sim A \vee \sim B) \wedge (A \wedge B))$. First, consider how we might rewrite this proof using *DeMorgans*, in addition to the basic rules of SD, but only applying it to the whole sentence on a line.

33.

1.	$(\sim A \vee \sim B) \wedge (A \wedge B)$	*Assume*
2.	$(\sim A \vee \sim B)$	$\wedge$-*Elim*, 1
3.	$(A \wedge B)$	$\wedge$-*Elim*, 1
4.	$\sim(A \wedge B)$	*DeM*, 2
5.	$(A \wedge B) \wedge \sim(A \wedge B)$	$\wedge$-*Intro*, 3,4
6.	$((\sim A \vee \sim B) \wedge (A \wedge B)) \rightarrow ((A \wedge B) \wedge \sim(A \wedge B))$	$\rightarrow$-*Intro*, 1–5
7.	$\sim((\sim A \vee \sim B) \wedge (A \wedge B))$	$\sim$-*Intro*, 6

Obviously this is much shorter than 26, and much shorter than 26 would be even if we rewrote it using *A.C.* (as we suggested the reader do above). Here we have applied *DeMorgans* to line 2. But if we allow ourselves to apply *DeMorgans* to subsentences on lines in addition to whole sentences, then the proof can be made even shorter.

34.

1.	$(\sim A \vee \sim B) \wedge (A \wedge B)$	*Assume*
2.	$\sim(A \wedge B) \wedge (A \wedge B)$	*DeM*, 1
3.	$((\sim A \vee \sim B) \wedge (A \wedge B)) \rightarrow (\sim(A \wedge B) \wedge (A \wedge B))$	$\rightarrow$-*Intro*, 1–2
4.	$\sim((\sim A \vee \sim B) \wedge (A \wedge B))$	$\sim$-*Intro*, 3

Unlike in 33, here we did not need to break line 1 apart into its conjuncts. We simply applied *DeMorgans* to the first conjunct of line 1 and wrote the result down as line 2. To see how definition 6.9 (on the previous page) captures this intuitive idea, consider how we would show, directly from the definition, that *DeMorgans* sanctions writing $\psi^* = \sim(A \wedge B) \wedge (A \wedge B)$ on line 2 when applied to $\psi = (\sim A \vee \sim B) \wedge (A \wedge B)$ on line 1. The given schema for *DeMorgans* relevant to line 1 is $\sim\phi_1 \vee \sim\phi_2$, while the relevant may-add schema is $\sim(\phi_1 \wedge \phi_2)$. The substitution we want for clause (1) of definition 6.9 is $\phi_1 = A$ and $\phi_2 = B$. This substitution results in $\phi = (\sim A \vee \sim B)$ and $\phi^* = \sim(A \wedge B)$. In line with clause (2) of the definition, $\phi = (\sim A \vee \sim B)$ is a subsentence

of $\psi = (\sim A \vee \sim B) \wedge (A \wedge B)$. And, in line with clause (3), $\psi^* = \sim(A \wedge B) \wedge (A \wedge B)$ is the SL sentence you get when you replace one instance (token) of $\phi = (\sim A \vee \sim B)$ with an instance (token) of $\phi^* = \sim(A \wedge B)$ in $\psi = (\sim A \vee \sim B) \wedge (A \wedge B)$.

We have shown that the exchange shortcut rules, more liberally allowed to be applied to subsentences on a line, still have truth-preserving applications (Thm. 6.10). In the last section, 6.3.1 (on page 144), we showed that anything we can derive using the standard and exchange shortcut rules from SD$^+$ can be derived using only the basic rules of SD (Thm. 6.7, on page 148). But to prove this we used theorem 6.5 (on page 147) and theorem 6.6 and we need to think about how more liberally allowing the exchange shortcut rules to be applied to subsentences on a line affects the proofs of these two theorems. That is, if we want to show that theorem 6.7 still holds, we need to show that the theorems we used to prove it still hold.

It should not be hard to see that the change from definition 6.2 (on page 130) to 6.9 (on page 153), i.e. the move to more liberal applications of exchange shortcut rules, does not affect the proof of 6.5 (on page 147). But it does affect the proof of 6.6. We proved this theorem (or, rather, asked the reader to prove most of the cases) by giving, for each rule R, a derivation schema using only the basic rules of SD that has the given schemas of R as the premises and the may-add schema of R as the conclusion (as the last line). This was sufficient to show that every application of the rules of SD$^+$, if restricted to whole sentences on lines, is derivable from the basic rules of SD because if the rules are only being applied to whole sentences, then these derivation schema will yield derivations that use only basic rules for every application. But this will not work if we're more liberally allowing the exchange rules to be applied to subsentences on a line.

For example, consider derivation schema 32 (on page 151) for *DeMorgans*. Say that we have the sentence $A \vee (\sim B \wedge \sim C)$ on some line of a derivation on which we're working and we want to apply *DeMorgans* to the right disjunct. Under the more liberal definition 6.9 (on page 153) we can do this, and *DeMorgans* will permit, or sanction, us to write down $A \vee \sim(B \vee C)$. But it should be clear that substituting $\phi = B$ and $\theta = C$ into derivation schema 32 will not result in a derivation of $A \vee \sim(B \vee C)$ from $A \vee (\sim B \wedge \sim C)$.

To show that theorem 6.6 still holds for our more liberal use of exchange shortcut rules we rely on two facts. First, definition 6.9 ensures that if an exchange shortcut rule, applied to a sentence ψ, sanctions writing down ψ^*, then there's a specific relationship between ψ and ψ^*. Specifically, there are two sentences ϕ and ϕ^*, got by substituting sentences into the given and may-add schemas of the rules, and ψ^* is ψ with ϕ replaced with ϕ^*. The second fact is that these sentences ϕ and ϕ^* are always *provably equivalent*, or as we might say *derivationally equivalent*.[7] (This is proved in exercise 6.4.2, on page 160.)

[7] Compare this definition with definition 7.8 (on page 172), which generalizes it for formulas of QL.

Definition 6.12. Two sentences of SL are *provably equivalent* iff one of the following two equivalent conditions holds:

(1) both $\phi \vdash \psi$ and $\psi \vdash \phi$, or

(2) $\vdash \phi \leftrightarrow \psi$.

These two facts, along with the following theorem,[8] are enough to show that theorem 6.6 still holds for our more liberal use of exchange shortcut rules. (The reader should make sure they are convinced of that.)

Theorem 6.13. Restricted Replacement Theorem for SD: For all sentences ψ of SL: if

(1) ϕ and ϕ^* are SL sentences such that $\phi \vdash \phi^*$ and $\phi^* \vdash \phi$, and

(2) if ϕ is a subsentence of ψ, then ψ^* is the SL sentence you get when you replace one instance (token) of ϕ with an instance (token) of ϕ^*, and ψ^* is ψ if not,

then ψ^* can be derived from ψ using only the basic rules of SD, i.e. $\psi \vdash \psi^*$.

Proof: Assume that ϕ and ϕ^* are two SL sentences such that $\phi \vdash \phi^*$ and $\phi^* \vdash \phi$.

Base Step: If ψ is atomic, then it's just a sentence letter. So, if ϕ is a subsentence of ψ, it itself must just be that same sentence letter. So, ϕ is the same sentence as ψ, and ψ^* is the same as ϕ^*. Since $\phi \vdash \phi^*$, it follows immediately that $\psi \vdash \psi^*$.

Inheritance Step: For the recursive hypothesis, assume that the theorem holds for the sentences $\theta, \theta_1, \ldots, \theta_n, \delta$.

Conjunction: Assume that ψ is the conjunction $\theta_1 \wedge \ldots \wedge \theta_n$. Either ϕ is not a subsentence of ψ, it's a subsentence of ψ but not the same as ψ, or is the same as ψ. If it's not a subsentence of ψ or it's the same as ψ, then just as in the base step it follows immediately that $\psi \vdash \psi^*$.

So assume that ϕ is a subsentence of ψ but not the same as it. Then ϕ is a subsentence of one of the conjuncts θ_i of ψ and ψ^* is the conjunction $\theta_1 \wedge \ldots \wedge \theta_i^* \wedge \ldots \wedge \theta_n$. By the recursive hypothesis, $\theta_i \vdash \theta_i^*$. It should be clear that by using $\wedge$-*Elim* we have that $\psi \vdash \theta_1$, ..., $\psi \vdash \theta_i$, ..., $\psi \vdash \theta_n$. By transitivity, $\psi \vdash \theta_i^*$. And, it should be clear that if each of $\theta_1, \ldots, \theta_i^*, \ldots, \theta_n$ can be derived from ψ, then by using $\wedge$-*Intro* we can derive their conjunction $\theta_1 \wedge \ldots \wedge \theta_i^* \wedge \ldots \wedge \theta_n$ from ψ. But this conjunction just is ψ^*, so $\psi \vdash \psi^*$.

[8] Compare this theorem to theorem 7.6 (on page 172), which is a stronger version of it generalized to QD. Note that the Restricted Replacement Theorem for SD follows immediately from the generalized QD version (but we provide a separate proof here). Also note that the proof of theorem 7.6 uses the One-step Replacement Lemmas (Thm. 7.9, on page 173), and the proofs for these involve constructing derivation schemas. The proof given here for the Restricted Replacement Theorem for SD also involves the construction of derivation schemas, but does so by giving instructions in the inheritance step for how to write the relevant derivations instead of explicitly writing them out in a separate lemma.

Disjunction: Assume that ψ is the disjunction $\theta_1 \vee ... \vee \theta_n$. Either ϕ is not a subsentence of ψ, it's a subsentence of ψ but not the same as ψ, or is the same as ψ. If it's not a subsentence of ψ or it's the same as ψ, then just as in he base step it follows immediately that $\psi \vdash \psi^*$.

So assume that ϕ is a subsentence of ψ but not the same as it. Then ϕ is a subsentence of one of the disjuncts θ_i of ψ and ψ^* is the disjunction $\theta_1 \vee ... \vee \theta_i^* \vee ... \vee \theta_n$.

We want to show that $\psi \vdash \psi^*$, i.e. that $\theta_1 \vee ... \vee \theta_n \vdash \theta_1 \vee ... \vee \theta_i^* \vee ... \vee \theta_n$. Using the proof by contradiction strategy, we write ψ on the first line and write $\sim\psi^*$ on line 2 as an assumption with the goal of deriving a contradiction. Now we apply *DeMorgans* to line 2, getting $\sim\theta_1 \wedge ... \wedge \sim\theta_i^* \wedge ... \wedge \sim\theta_n$. Now using $\wedge$-*Elim* we break apart this conjunction, getting each conjunct $\sim\theta_1, ..., \sim\theta_i^*, ..., \sim\theta_n$ on a separate line. Then using these conjuncts and *Disjunctive Syllogism* on line 1 to get θ_i on its own line. By the recursive hypothesis, $\theta_i \vdash \theta_i^*$, so from the line with θ_i we can derive θ_i^*. Since we already have $\sim\theta_i^*$ on its own line, we can use $\wedge$-*Intro* to get $\theta_i^* \wedge \sim\theta_i^*$. We then close the assumption on line 2 by using $\rightarrow$-*Intro* to get $\sim\psi^* \rightarrow (\theta_i^* \wedge \sim\theta_i^*)$. We finally use $\sim$-*Elim* to get ψ^*.

Negation: Assume that ψ is the negation $\sim\theta$. Either ϕ is not a subsentence of ψ, it's a subsentence of ψ but not the same as ψ, or is the same as ψ. If it's not a subsentence of ψ or it's the same as ψ, then just as in he base step it follows immediately that $\psi \vdash \psi^*$.

So assume that ϕ is a subsentence of ψ but not the same as it. Then ϕ is a subsentence of θ and ψ^* is the negation $\sim\theta^*$.

We want to show that $\psi \vdash \psi^*$, i.e. that $\sim\theta \vdash \sim\theta^*$. Using our usual basic bottom-up strategy for negation, we set up this proof by putting $\sim\theta$ on the first line and assuming θ^* with the goal of deriving a contradiction. But, by the recursive hypothesis, $\theta^* \vdash \theta$. So we know that from assumption θ^* we can derive θ, and at that point we have $\sim\theta$ on one line and θ on another. Using $\wedge$-*Intro* we can get $\theta \wedge \sim\theta$. Then with $\rightarrow$-*Intro* we get $\theta^* \rightarrow (\theta \wedge \sim\theta)$, and applying $\sim$-*Intro* to this sentence will get us $\sim\theta^*$.

Conditional: Left to the reader as an exercise.

Biconditional: Also left to the reader as an exercise.

Closure Step: Since the inheritance step covers all the ways to generate SL sentences, we've shown that the theorem holds for all SL sentences ψ.

■

Thus we have shown that anything we can derive in SD^+ can be derived in SD.

6.3.3 Shortcut Rule Strategies

As we discussed in section 6.2.6, for each logical connective there are two types of strategies: those for what to do if you already have sentences with that as their main connective (top-down strategies), and those for what to do if you want to get a sentence with that as its main connective (bottom-up strategies). In section 6.2.6 we covered basic top-down and bottom-up strategies for each connective. We now add new strategies based on shortcut rules to this basic stock. (Since shortcut rules don't divide nicely by connective, and often involve schemas with multiple connectives, some of the groupings here a bit arbitrary; but this isn't a substantial issue.)

Conjunction

***DeM* Top-down:** If you have a sentence of the form $\sim\phi_1 \wedge ... \wedge \sim\phi_n$, then convert it using *DeM* to get $\sim(\phi_1 \vee ... \vee \phi_n)$ on a new line.

$\sim/\rightarrow$-*Exchange* Top-down: If you have a sentence of the form $\phi \wedge \sim\psi$, then convert it using $\sim/\rightarrow$-*Exchange* to get $\sim(\phi \rightarrow \psi)$ on a new line.

Disjunction

***D.S.* Top-down:** If you have a sentence of the form $\phi_1 \vee ... \vee \phi_i \vee ... \vee \phi_n$, and another sentence of the form $\sim\phi_i$, then eliminate one of the disjuncts using *D.S.* to get $\phi_1 \vee ... \vee \phi_{i-1} \vee \phi_{i+1} \vee ... \vee \phi_n$ on a new line.

***DeM* Top-down:** If you have a sentence of the form $\sim\phi_1 \vee ... \vee \sim\phi_n$, then convert it using *DeM* to get $\sim(\phi_1 \wedge ... \wedge \phi_n)$ on a new line.

$\rightarrow/\sim$-*Exchange* Top-down: If you have a sentence of the form $\sim\phi \vee \psi$, then convert it using $\rightarrow/\sim$-*Exchange* to get $\phi \rightarrow \psi$ on a new line.

Negation

***DeM* Top-down:** If you have a sentence of the form $\sim(\phi_1 \wedge ... \wedge \phi_n)$, then convert it using *DeM* to get $\sim\phi_1 \vee ... \vee \sim\phi_n$ on a new line.

***DeM* Top-down:** If you have a sentence of the form $\sim(\phi_1 \vee ... \vee \phi_n)$, then convert it using *DeM* to get $\sim\phi_1 \wedge ... \wedge \sim\phi_n$ on a new line.

$\sim\sim$-*Elim* Top-down: If you have a sentence of the form $\sim\sim\phi$, then convert it using $\sim\sim$-*Elim* to get ϕ on a new line.

$\sim/\rightarrow$-*Exchange* Top-down: If you have a sentence of the form $\sim(\phi \rightarrow \psi)$, then convert it using $\sim/\rightarrow$-*Exchange* to get $\phi \wedge \sim\psi$ on a new line.

$\sim\sim$-*Intro* Bottom-up: If you want to get a sentence of the form $\sim\sim\phi$, then first derive ϕ and then use $\sim\sim$-*Intro* to derive it.

Conditionals

***M.T.* Top-down:** If you have a sentence of the form $\phi \to \psi$, and another $\sim\psi$, then break the conditional apart using *M.T.* to get $\sim\phi$ on a new line.

$\to /\sim$-*Exchange* Top-down: If you have a sentence of the form $\phi \to \psi$, then convert it using $\to /\sim$-*Exchange* to get $\sim\phi \vee \psi$ on a new line.

***Contraposition* Top-down:** If you have a sentence of the form $\phi \to \psi$, then convert it using *Contraposition* to get $\sim\psi \to \sim\phi$ on a new line.

Biconditionals

$\sim/\leftrightarrow$-*Intro* Top-down: If you have a sentence of the form $\phi \leftrightarrow \psi$, then convert it using $\sim/\leftrightarrow$-*Intro* to get $\sim\phi \leftrightarrow \sim\psi$ on a new line.

Misc

***A.C.* Bottom-up:** If you want to get a sentence of the form ψ, then first derive both ϕ and $\sim\phi$ and then use *A.C.* to derive it from them.

Note that we've left strategies derived from *Distribution*. This isn't because they're not useful (quite the contrary). Instead, we leave it to the reader to place the appropriate Top-down strategies from *Distribution* under conjunction and disjunction.

Also note that we've presented most of the strategies as top-down. For any of the strategies based on an exchange shortcut rule, it should be clear that you can "reverse" the strategy and read it as bottom-up.

6.4 Exercises

6.4.1 SD Practice Problems

Write derivations for each of the following using only the rules specified by the instructor. It is probably a good idea to do the problems in order, as the earlier ones tend to be easier than the later ones.

1. $\vdash A \to (B \to A)$
2. $\vdash (A \to B) \to ((B \to C) \to (A \to C))$
3. $\vdash (A \to B) \to ((A \vee C) \to (B \vee C \vee D))$
4. $\vdash ((A \to B) \wedge C \wedge (C \leftrightarrow A)) \to B$
5. $\vdash \sim A \vee (B \to A)$
6. $\vdash (A \to B) \vee (B \to C)$
7. $\vdash (\sim C \wedge ((A \to C) \vee (B \to C))) \to \sim(A \wedge B)$

6.4.2 SD Shortcut Rules

Finish the proof of theorem 6.6 (on page 148). To do this, write derivation schemas for each of the following using only the basic rules of SD or shortcut rules for which you've already done a derivation schema. Note that this is also sufficient to show that sentences got by substituting into the given and may-add schemas of the SD exchange shortcut rules are provably equivalent (Def. 6.12, on page 156). Note that a star * has been placed next to the ones that have already been done in the text above. (They are left in the list for completeness.) Hint: derivations 16 and 25 should be helpful in doing two of the problems below.

M.T.

1. $\phi \rightarrow \theta, \sim\theta \vdash \sim\phi$

D.S.

2. $\phi \vee \theta, \sim\phi \vdash \theta$
3. $\sim\phi \vee \theta, \phi \vdash \theta$

A.C.

4. $\phi, \sim\phi \vdash \psi$ * (derivation 28)

$\sim/\leftrightarrow$-Intro

5. $\phi \leftrightarrow \psi \vdash \sim\phi \leftrightarrow \sim\psi$

DeM

6. $\sim(\phi \wedge \theta) \vdash (\sim\phi \vee \sim\theta)$
7. $\sim\phi \vee \sim\theta \vdash \sim(\phi \wedge \theta)$
8. $\sim(\phi \vee \theta) \vdash (\sim\phi \wedge \sim\theta)$
9. $\sim\phi \wedge \sim\theta \vdash \sim(\phi \vee \theta)$ * (derivation 32)

Distribution

18. $\theta \wedge (\phi_1 \vee \phi_2) \vdash (\theta \wedge \phi_1) \vee (\theta \wedge \phi_2)$
19. $(\theta \wedge \phi_1) \vee (\theta \wedge \phi_2) \vdash \theta \wedge (\phi_1 \vee \phi_2)$
20. $(\phi_1 \vee \phi_2) \wedge \theta \vdash (\phi_1 \wedge \theta) \vee (\phi_2 \wedge \theta)$
21. $(\phi_1 \wedge \theta) \vee (\phi_2 \wedge \theta) \vdash (\phi_1 \vee \phi_2) \wedge \theta$
22. $\theta \vee (\phi_1 \wedge \phi_2) \vdash (\theta \vee \phi_1) \wedge (\theta \vee \phi_2)$
23. $(\theta \vee \phi_1) \wedge (\theta \vee \phi_2) \vdash \theta \vee (\phi_1 \wedge \phi_2)$
24. $(\phi_1 \wedge \phi_2) \vee \theta \vdash (\phi_1 \vee \theta) \wedge (\phi_2 \vee \theta)$

$\sim\sim$-Elim

10. $\sim\sim\phi \vdash \phi$

$\sim\sim$-Intro

11. $\phi \vdash \sim\sim\phi$

$\rightarrow / \vee$-Exchange

12. $\phi \rightarrow \theta \vdash \sim\phi \vee \theta$
13. $\sim\phi \vee \theta \vdash \phi \rightarrow \theta$

Contraposition

14. $\phi \rightarrow \theta \vdash \sim\theta \rightarrow \sim\phi$
15. $\sim\theta \rightarrow \sim\phi \vdash \phi \rightarrow \theta$

$\sim/ \rightarrow$-Exchange

16. $\sim(\phi \rightarrow \theta) \vdash \phi \wedge \sim\theta$
17. $\phi \wedge \sim\theta \vdash \sim(\phi \rightarrow \theta)$

25. $(\phi_1 \vee \theta) \wedge (\phi_2 \vee \theta) \vdash (\phi_1 \wedge \phi_2) \vee \theta$
26. $\theta \leftrightarrow \psi \vdash (\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)$
27. $(\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi) \vdash \theta \leftrightarrow \psi$

Chapter 7

Quantificational Derivations

7.1 The Basic System QD

Now our goal is to extend SD to a natural deduction system for QL. This system, which we call *Quantificational Derivation System*, or QD, consists of all the rules of SD, plus an introduction and elimination rule for each quantifier (given in table 7.1). Note that some of the rules for QD have special restrictions. We explain these in the examples below.

Name	**Given**	**May Add**
$\forall$-*Elim*	$\forall\beta\phi$	$\phi\mathrm{a}/\beta$, for 'a' any individual constant
$\forall$-*Intro*	$\phi\mathrm{a}/\beta$	$\forall\beta\phi$, iff 'a' does not occur in ϕ nor in any unboxed line justified by Assumption
$\exists$-*Intro*	$\phi\mathrm{a}/\beta$	$\exists\beta\phi$
$\exists$-*Elim*	$\exists\beta\phi$, $\phi\mathrm{a}/\beta\rightarrow\theta$	θ, iff 'a' does not occur in ϕ or θ, nor in any unboxed line justified by Assumption

Table 7.1: (New) Basic Rules for QD

The mechanics of writing proofs in QD are no different than the mechanics of writing proofs in SD, but we do have to slightly adapt the definition for sanctioning (Def. 6.2, on page 130).

Definition 7.1. A rule R, applied to unboxed lines $m_1, \ldots, m_j$ with, respectively, sentences $\psi_1, \ldots, \psi_j$, *sanctions* writing the sentence ϕ iff there's some substitution of QL *formulas*

that, for the given schemas of R, results in $\psi_1, \ldots, \psi_j$ and, for the may-add schema, results in ϕ.

Another difference is that there are four new rules available. We now look at four examples which highlight each rule.

Our first example demonstrates $\forall$-*Elim*. Say we want to show that $\forall x(Ax \rightarrow Bx), A\text{d} \vdash B\text{d}$. We start by setting the two sentences on the LHS of the turnstile as assumptions:

35.

1.	$\forall x(Ax \rightarrow Bx)$	*Assume*
2.	$\quad A\text{d}$	*Assume*

Next we use $\forall$-*Elim* on line 1. Note that there are no restrictions on the use of $\forall$-*Elim*.

36.

1.	$\forall x(Ax \rightarrow Bx)$	*Assume*
2.	$\quad A\text{d}$	*Assume*
3.	$\quad A\text{d} \rightarrow B\text{d}$	$\forall$-*Elim*, 1

Although we could have substituted any constant for x in line 3, we chose d so that we can next apply $\rightarrow$-*Elim*.

37.

1.	$\forall x(Ax \rightarrow Bx)$	*Assume*
2.	$\quad A\text{d}$	*Assume*
3.	$\quad A\text{d} \rightarrow B\text{d}$	$\forall$-*Elim*, 1
4.	$\quad B\text{d}$	$\rightarrow$-*Elim*, 2,3

And this completes the derivation.

The next example demonstrates $\exists$-*Intro*. Here we show that $\forall y(\exists x Dxy \rightarrow By), D\text{ab} \vdash B\text{b}$. As before, we start with the assumptions.

38.

1.	$\forall y(\exists x Dxy \rightarrow By)$	*Assume*
2.	$\quad D\text{ab}$	*Assume*

Again as before, we need to use $\forall$-*Elim* so we can eventually use $\rightarrow$-*Elim* to finish.

39.

1.	$\forall y(\exists x Dxy \rightarrow By)$	*Assume*
2.	$\quad D\text{ab}$	*Assume*
3.	$\quad \exists x Dx\text{b} \rightarrow B\text{b}$	$\forall$-*Elim*, 1

Again we strategically chose the constant we did, b, so that using $\rightarrow$-*Elim* will get us the right result. But we can't apply $\rightarrow$-*Elim* yet, since the LHS of the horseshoe in line 3, $\exists x Dx\text{b}$, is an existential sentence, while what we have on line 2, $D\text{ab}$, is not. But, by using $\exists$-*Intro* we can easily get what we need:

40.

1.	$\forall y(\exists x Dxy \rightarrow By)$	*Assume*
2.	$\quad Da\mathrm{b}$	*Assume*
3.	$\quad \exists x Dx\mathrm{b} \rightarrow B\mathrm{b}$	$\forall$-*Elim*, 1
4.	$\quad \exists x Dx\mathrm{b}$	$\exists$-*Intro*, 2

Note that just as with $\forall$-*Elim* there are no restrictions on $\rightarrow$-*Elim*. Also note that we could have used $\rightarrow$-*Elim* to generalize the constant b, getting $\exists x Da x$, but that wouldn't have helped us. Also, we could have generalized using a different variable, getting, say $\exists z Dz\mathrm{b}$. But again that wouldn't have helped us. Any constant is legal to instantiate with $\forall$-*Elim*, but often only one is a wise choice.

We now have the right setup for $\rightarrow$-*Elim*:

41.

1.	$\forall y(\exists x Dxy \rightarrow By)$	*Assume*
2.	$\quad Da\mathrm{b}$	*Assume*
3.	$\quad \exists x Dx\mathrm{b} \rightarrow B\mathrm{b}$	$\forall$-*Elim*, 1
4.	$\quad \exists x Dx\mathrm{b}$	$\exists$-*Intro*, 2
5.	$\quad B\mathrm{b}$	$\rightarrow$-*Elim*, 3,4

And this completes the derivation.

The next example demonstrates $\exists$-*Elim*. This is our first rule with restrictions. We show that $\forall x{\sim}Qx\mathrm{b}, \exists z(Qz\mathrm{b} \vee Gz) \vdash \exists x Gx$. As before, we start by setting out the assumptions.

42.

1.	$\forall x{\sim}Qx\mathrm{b}$	*Assume*
2.	$\quad \exists z(Qz\mathrm{b} \vee Gz)$	*Assume*

When using both $\exists$-*Elim* and $\forall$-*Elim*, it's generally best to take the instance of the existential for $\exists$-*Elim* first and use $\forall$-*Elim* after. In some cases you want the $\forall$-*Elim* instance to match the $\exists$-*Elim* constant, but in other cases you want it to be different. Now, according to $\exists$-*Elim*, we can get a sentence θ in this case by showing that $(Qt\mathrm{b} \vee Gt) \rightarrow \theta$, for some constant t that fits the restriction on $\exists$-*Elim*. (Just what constants will work and what sentence θ we want will take some thought.) So, we need to use $\rightarrow$-*Intro*.

43.

1.	$\forall x{\sim}Qx\mathrm{b}$	*Assume*
2.	$\quad\exists z(Qz\mathrm{b} \vee Gz)$	*Assume*
3.	$\quad\boxed{\begin{array}{l}Q\mathrm{ab} \vee G\mathrm{a}\\ \\ \vdots \\ \\ \theta\end{array}}$	*Assume*
n.		
$n+1$.	$\quad(Q\mathrm{ab} \vee G\mathrm{a}) \rightarrow \theta$	$\rightarrow$-*Intro*, 3–n
$n+2$.	$\quad\theta$	$\exists$-*Elim*, 2,$n+1$

Here we have chosen a to substitute for z in line 3 because (so long as we pick θ correctly) it will meet our restriction. According to the restriction on $\exists$-*Elim*, the constant we pick can't appear in any open assumptions (the assumption used to derive the needed conditional, in this case line 3, is not open by the time we apply the rule). It also can't appear in the scope of the existential quantifier, which in this case is $(Qz\mathrm{b} \vee Gz)$. Since a does not appear in any open assumptions and does not appear in $(Qz\mathrm{b} \vee Gz)$, it will meet our restriction. (Clearly other constants would have also met the restriction.) Now we have to decide what sentence θ enables us to finish the derivation. Note that if we derive Gt for any constant $t \neq \mathrm{a}$, then we can use $\exists$-*Intro* to get the needed sentence $\exists xGx$. (We will set this up, but we see in a moment that this first guess won't work.) We can't pick $t = \mathrm{a}$ because then we couldn't apply $\exists$-*Elim* on line $n+2$.

44.

1.	$\forall x{\sim}Qx\mathrm{b}$	*Assume*
2.	$\quad\exists z(Qz\mathrm{b} \vee Gz)$	*Assume*
3.	$\quad\boxed{\begin{array}{l}Q\mathrm{ab} \vee G\mathrm{a}\\ \\ \vdots \\ \\ G\mathrm{c}\end{array}}$	*Assume*
n.		
$n+1$.	$\quad(Q\mathrm{ab} \vee G\mathrm{a}) \rightarrow G\mathrm{c}$	$\rightarrow$-*Intro*, 3–n
$n+2$.	$\quad G\mathrm{c}$	$\exists$-*Elim*, 2,$n+1$
$n+3$.	$\quad\exists xGx$	$\exists$-*Intro*, $n+2$

Now we only need to complete the derivation by deriving $G\mathrm{c}$ from $(Q\mathrm{ab} \vee G\mathrm{a})$. We can *try* do this by using $\forall$-*Elim* and *D.S.*, but it becomes clear at once that this won't work.

45.

1.	$\forall x{\sim}Qx\mathrm{b}$	*Assume*
2.	$\quad\exists z(Qz\mathrm{b} \vee Gz)$	*Assume*
3.	$\quad\boxed{Q\mathrm{ab} \vee G\mathrm{a}}$	*Assume*
4.	$\quad\boxed{{\sim}Q\mathrm{ab}}$	$\forall$-*Elim*, 1
5	$\quad\boxed{G\mathrm{a}}$	*D.S.*, 3,4
	$\quad\boxed{\vdots}$	
n.	$\quad\boxed{G\mathrm{c}}$	
$n+1$.	$\quad(Q\mathrm{ab} \vee G\mathrm{a}) \rightarrow G\mathrm{c}$	$\rightarrow$-*Intro*, 3–n
$n+2$.	$\quad G\mathrm{c}$	$\exists$-*Elim*, 2,$n+1$
$n+3$.	$\quad\exists xGx$	$\exists$-*Intro*, $n+2$

It should be clear that this $\forall$-*Elim*/*D.S.* strategy won't work, since for it to work we'd need the token of G that appears in line 3 to be followed by the *same* constant that follows the token of G that appears in line n. But that constant is the constant we substitute in for $\exists$-*Elim*, and we would then violate the restriction for the rule.

So we need to go back to θ and consider another strategy. This time we try letting $\theta = \exists xGx$.

46.

1.	$\forall x{\sim}Qx\mathrm{b}$	*Assume*
2.	$\quad\exists z(Qz\mathrm{b} \vee Gz)$	*Assume*
3.	$\quad\boxed{Q\mathrm{ab} \vee G\mathrm{a}}$	*Assume*
	$\quad\boxed{\vdots}$	
n.	$\quad\boxed{\exists xGx}$	
$n+1$.	$\quad(Q\mathrm{ab} \vee G\mathrm{a}) \rightarrow \exists xGx$	$\rightarrow$-*Intro*, 3–n
$n+2$.	$\quad\exists xGx$	$\exists$-*Elim*, 2,$n+1$

Note that we still keep our choice of a on line 3 within the restrictions of $\exists$-*Elim*. Now we can try again at finishing the proof. This time we can:

47.

1.	$\forall x{\sim}Qx\mathrm{b}$	*Assume*
2.	$\quad\exists z(Qz\mathrm{b} \vee Gz)$	*Assume*
3.	$\quad\boxed{Q\mathrm{ab} \vee G\mathrm{a}}$	*Assume*
4.	$\quad\boxed{{\sim}Q\mathrm{ab}}$	$\forall$-*Elim*, 1
5.	$\quad\boxed{G\mathrm{a}}$	*D.S.*, 3,4
6.	$\quad\boxed{\exists xGx}$	$\exists$-*Intro*, 5
7.	$\quad(Q\mathrm{ab} \vee G\mathrm{a}) \rightarrow \exists xGx$	$\rightarrow$-*Intro*, 3–6
8.	$\quad\exists xGx$	$\exists$-*Elim*, 2,7

And this completes the derivation.

The final example demonstrates $\forall$-*Intro*. This rule also has restrictions. We show that $\forall x(Ax \rightarrow Bx), \forall x(Bx \rightarrow Hx) \vdash \forall x(Ax \rightarrow Hx)$. It's worth mentioning that the fact that all the examples so far have two sentences on the LHS of the turnstile is just a coincidence. As before, we start by setting out the assumptions.

48.

1.	$\forall x(Ax \rightarrow Bx)$	*Assume*
2.	$\quad\forall x(Bx \rightarrow Hx)$	*Assume*

Before moving forward, we need to think about how $\forall$-*Intro* works. According to the rule, if we want to get $\forall x(Ax \rightarrow Hx)$ by using $\forall$-*Intro*, we need to first get $At \rightarrow Ht$ on some line, where t is some constant that does not appear in any unboxed assumptions, or in $Ax \rightarrow Hx$. Since no constants appear in $Ax \rightarrow Hx$, it provides no constraints. Further, there are no constants in either unboxed assumption in derivation 48, so we are free to chose any constant we like. Pick a. So we want to derive $A\mathrm{a} \rightarrow H\mathrm{a}$.

49.

1.	$\forall x(Ax \rightarrow Bx)$	*Assume*
2.	$\quad\forall x(Bx \rightarrow Hx)$	*Assume*
3.	$\quad\boxed{A\mathrm{a}}$	*Assume*
	$\quad\boxed{\vdots}$	
n.	$\quad\boxed{H\mathrm{a}}$	
$n+1$.	$\quad A\mathrm{a} \rightarrow H\mathrm{a}$	$\rightarrow$-*Intro*, 3–n
$n+2$.	$\quad\forall x(Ax \rightarrow Hx)$	$\forall$-*Intro*, $n+1$

Now we just have to finish the derivation of $H\mathrm{a}$ from $A\mathrm{a}$ without introducing any new unboxed assumptions with constant a. (Of course, if we did the bottom half of this proof wouldn't work, since we coudn't close the assumption on line 3 with $\rightarrow$-*Intro* if unboxed assumptions appeared after line 3.) A natural next step is to use $\forall$-*Elim* on lines 1 and 2.

50.

1.	$\forall x(Ax \to Bx)$	*Assume*
2.	$\quad \forall x(Bx \to Hx)$	*Assume*
3.	$\quad\quad \boxed{Aa}$	*Assume*
4.	$\quad\quad \boxed{Aa \to Ba}$	$\forall$-*Elim*, 1
5.	$\quad\quad \boxed{Ba \to Ha}$	$\forall$-*Elim*, 2
	$\quad\quad \boxed{\vdots}$	
n.	$\quad\quad \boxed{Ha}$	
$n+1$.	$\quad Aa \to Ha$	$\to$-*Intro*, 3–n
$n+2$.	$\quad \forall x(Ax \to Hx)$	$\forall$-*Intro*, $n+1$

Now getting to Ha is just a matter of using $\to$-*Elim*:

51.

1.	$\forall x(Ax \to Bx)$	*Assume*
2.	$\quad \forall x(Bx \to Hx)$	*Assume*
3.	$\quad\quad \boxed{Aa}$	*Assume*
4.	$\quad\quad \boxed{Aa \to Ba}$	$\forall$-*Elim*, 1
5.	$\quad\quad \boxed{Ba \to Ha}$	$\forall$-*Elim*, 2
6.	$\quad\quad \boxed{Ba}$	$\to$-*Elim*, 3,4
7.	$\quad\quad \boxed{Ha}$	$\to$-*Elim*, 5,6
8.	$\quad Aa \to Ha$	$\to$-*Intro*, 3–7
9.	$\quad \forall x(Ax \to Hx)$	$\forall$-*Intro*, 8

And this completes the derivation. Notice that since there are infinitely many constants in QL, and any of them would have worked in the derivation, there are infinitely many derivations of this sentence.

7.1.1 Decidability

There are multiple algorithms to follow for applying the rules which, if there does exist a derivation, will halt when the last line written down is the sentence to be derived. For SD the algorithms are intuitive and straightforward (see Sec. 8.3, on page 185), while for QD (the basic derivation system for QL we define in Sec. 7.1, on page 163) they are much more complicated.

But although these algorithms are guaranteed to end in a derivation if the sentence can be derived, there are at least two reasons why you don't want to do most of your derivations using them. First, the derivations produced by them tend to be much

longer and more complicated than is necessary. You will almost always be able to come up with a much shorter and more direct proof on your own. Second, at least for QD (but not SD), although *if* there is a derivation the algorithms will "find it", if there is *not* a derivation then the algorithms may never "find out". That is, if there is not a derivation then the algorithms (any one you pick) do essentially one of two things: either they halt in a way that indicates there is no derivation, or they never halt. If you happen to be working on a problem in the latter case and you're only following the algorithm, then you'll never find out whether there is a derivation. If a derivation system, like QD, has this feature, then it's said to be *undecidable*. If there is an algorithm that always halts either in a derivation or with an indication that there's no derivation (as in the case of SD), then the system is said to be *decidable* and the algorithm is said to be a *decision procedure*. QD is undecidable, while SD and QD1 are decidable.[1] We will prove QD1 decidability in section 8.6, on page 206.

7.2 Shortcut Rules for QD

Like SD, we extend QD by adding shortcut rules. First, we include all the shortcut rules we gave for SD, both the standard and exchange rules. Second, we include shortcut rules specifically for the quantifiers (see table 7.19). These new *quantifier negation* rules are found in table 7.19. The system with all the shortcut rules from SD (tables 6.39 and 6.39) and the new shortcut rules for the quantifiers (table 7.19) is called QD$^+$.

Name	**Given**	**May Add**
QN	$\sim\forall\beta\phi$	$\exists\beta\sim\phi$
	$\exists\beta\sim\phi$	$\sim\forall\beta\phi$
	$\sim\exists\beta\phi$	$\forall\beta\sim\phi$
	$\forall\beta\sim\phi$	$\sim\exists\beta\phi$

Table 7.19: Exchange Short-Cut Rules for QD

Just as we had to slightly modify the definition of sanctioning used in SD for the basic (intro and elimination) rules and standard shortcut rules for QD, we also have to slightly modify the definition of sanctioning used in SD$^+$ (Def. 6.9, on page 153) for the exchange shortcut rules for QD that make up QD$^+$.

[1] Enderton (2010) provides a general, contemporary introduction to computability and decision procedures. Kleene (2002 [1967]: ch. 5) provides a lucid and concise discussion within roughly the framework devolved here.

Definition 7.2. An exchange shortcut rule R of QD$^+$ (a rule from table 6.40, on page 146, or table 7.19, on the previous page), applied to a line with a QL formula ψ, *sanctions* writing down sentence ψ^* iff

(1) there is some substitution of QL formulas that, for the given schema of R, results in a formula ϕ and, for the may-add schema, results in a formula ϕ^*,

(2) ϕ is a subformula of ψ, and

(3) ψ^* is the SL sentence you get when you replace one instance (token) of ϕ with an instance (token) of ϕ^* in ψ.

7.2.1 Shortcut Rule Elimination Theorem for QD

In this section we want to extend the Shortcut Rule Elimination Theorem (Thm. 6.7, on page 148) to QD.

Theorem 7.3. Shortcut Rule Elimination Theorem for QD$^+$: For all QL sentences $\phi_1, \dots, \phi_m$ and ψ, if ψ can be derived from $\phi_1, \dots, \phi_m$ in QD$^+$, then ψ can be derived from $\phi_1, \dots, \phi_m$ in QD.

The same proof we used for Shortcut Rule Elimination Theorem for SD (Thm. 6.7) will work for this version, so long as appropriate versions of theorems 6.5 (on page 147) and 6.6 (on page 148) hold for the shortcut rules of QD. Specifically:

Theorem 7.4. For all QL sentences $\theta_1, \dots, \theta_n, \delta$ and rules $R_1, \dots, R_p$, if

(1) δ can be derived from $\theta_1, \dots, \theta_n$ using rules $R_1, \dots, R_p$ and the basic rules of SD, and

(2) every application of a rule R_1 is derivable using the rules R_2, ..., R_p and the basic rules of QD (recall Def. 6.4, on page 147),

then δ can be derived from $\theta_1, \dots, \theta_n$ using only rules $R_2, \dots, R_p$ and the basic rules of QD.

Theorem 7.5. For all standard and exchange shortcut rules R (see tables 6.39, 6.40, and 7.19), every application of R is derivable using the basic rules of QD (see tables 6.2 and 7.1).

We leave it to the reader to prove the Shortcut Rule Elimination Theorem (Thm. 7.3) using theorems 7.4 and 7.5.

Turning to the proofs for theorems 7.4 and 7.5, note that nothing in the proof of 6.5 depended on any special features of SD. Thus, the proof of theorem 6.5 can be adapted to 7.4 just by changing all the references to SD to references to QD. But unfortunately theorem 7.5 is not nearly as straightforward.

It is helpful to break theorem 7.5 into two parts: (i) the claim that, for all standard shortcut rules (table 6.39), every application is derivable using the basic rules of QD,

and (ii) the claim that, for all the exchange shortcut rules (tables 6.40 and 7.19), every application is derivable using the basic rules of QD. Proving part (i) of theorem 7.5 is no different from proving it for theorem 6.6; the same arguments using the derivation schemas written for theorem 6.6 will work. But nothing done so far will help with part (ii). This is because applications of exchange shortcut rules in QD^+, even those shared with SD^+ (see table 6.40), use a different definition of sanctioning than was used in SD^+ (compare Def. 6.9 and 7.2). According to definition 7.2 (on the previous page), in QD^+ exchange rules can be applied not only to subsentences, but also to subformulas. We have to show that allowing exchange rules to be applied not only to subsentences, but also to subformulas, doesn't prevent us from deriving their applications using only the basic rules.

To do this we use (1) an extended (and generalized) version of the Restricted Replacement Theorem for SD (Thm. 6.13, on page 156) and (2) the fact that any two formulas got by substituting QL formulas into the may-add and given schemas of the exchange shortcut rules for QD are provably equivalent. (We ask the reader to prove this second fact in exercise 7.3.1, on page 176.)

Theorem 7.6. The Replacement Theorem for QD: If ϕ and ϕ^* are provably equivalent formulas of QL, and θ and θ^* differ only in that θ contains the subformula ϕ in one place where θ^* contains the subformula ϕ^*, then θ and θ^* are provably equivalent.

Before proving this theorem, we need to define when two *formulas* of QL are provably equivalent. The definition given for SL sentences (def. 6.12, on page 156) will not carry over to QL formulas, since formulas just aren't the sort of thing which can be derived. For example, we would like to be able to say that $(Qx \rightarrow Gy)$ and $(\sim Qx \vee Gy)$ are provably equivalent even though we cannot derive the formula $(Qx \rightarrow Gy) \leftrightarrow (\sim Qx \vee Gy)$ (because it's not a sentence).

The way we extend the notion of provable equivalence is through the universal closure of a formula.

Definition 7.7. The *universal closure* of a formula θ, written $\forall\theta$, is the sentence that results by prefixing universal quantifiers in alphabetical order for all free variables of θ.

E.g., $\forall((Qx \rightarrow Gy) \leftrightarrow (\sim Qx \vee Gy))$, the universal closure of $((Qx \rightarrow Gy) \leftrightarrow (\sim Qx \vee Gy))$, is $\forall x \forall y((Qx \rightarrow Gy) \leftrightarrow (\sim Qx \vee Gy))$.

We can now define provable equivalence for QL formulas using the universal closure:

Definition 7.8. Two QL formulas θ and ϕ are *provably equivalent* iff the universal closure of the formula that has a biconditional as main connective and θ and ϕ as immediate constituents is derivable in QD; in other words, iff $\vdash \forall(\theta \leftrightarrow \phi)$ in QD.

It's important to note that this definition really is a generalization of definition 6.12 (on page 156). If θ and ϕ are sentences of SL (recall that every sentence of SL is also a formula of QL), then they are provably equivalent on definition 6.12 iff they are provably equivalent on definition 7.8.

Before moving to the proof of the Replacement Theorem for QD (Thm. 7.6 on the previous page), it is convenient to prove the following one-step Replacement Lemmas:

Theorem 7.9. One-step Replacement Lemmas: If $\vdash \forall(\phi \leftrightarrow \phi^*)$, then:

(1) $\vdash \forall(\sim\phi \leftrightarrow \sim\phi^*)$

(2) $\vdash \forall\big((\phi \wedge \phi_1 \wedge ... \wedge \phi_p) \leftrightarrow (\phi^* \wedge \phi_1 \wedge ... \wedge \phi_p)\big)$

$\vdots$

(3) $\vdash \forall\big((\phi_1 \wedge ... \wedge \phi_p \wedge \phi) \leftrightarrow (\phi_1 \wedge ... \wedge \phi_p \wedge \phi^*)\big)$

(4) $\vdash \forall\big((\phi \vee \phi_1 \vee ... \vee \phi_p) \leftrightarrow (\phi^* \vee \phi_1 \vee ... \vee \phi_p)\big)$

$\vdots$

(5) $\vdash \forall\big((\phi_1 \vee ... \vee \phi_p \vee \phi) \leftrightarrow (\phi_1 \vee ... \vee \phi_p \vee \phi^*)\big)$

(6) $\vdash \forall((\phi \rightarrow \psi) \leftrightarrow (\phi^* \rightarrow \psi))$

(7) $\vdash \forall((\psi \rightarrow \phi) \leftrightarrow (\psi \rightarrow \phi^*))$

(8) $\vdash \forall((\phi \leftrightarrow \psi) \leftrightarrow (\phi^* \leftrightarrow \psi))$

(9) $\vdash \forall((\psi \leftrightarrow \phi) \leftrightarrow (\psi \leftrightarrow \phi^*))$

And, if $\vdash \forall\forall\beta(\phi \leftrightarrow \phi^*)$, then:

(10) $\vdash \forall(\forall\beta\phi \leftrightarrow \forall\beta\phi^*)$

(11) $\vdash \forall(\exists\beta\phi \leftrightarrow \exists\beta\phi^*)$

We prove (2) for the case of a 2-place conjunction and leave the rest to the reader to prove in a similar way. We use the following notation. If ϕ is a QL formula, then let $x_1, ..., x_m$ be the complete list of free variables in ϕ. Further, let $\phi c_1 ... c_m/x_1 ... x_m$ be the formula you get by substituting c_1 for x_1, ..., and c_m for x_m.

Proof of Thm. 7.9, (2), for 2-place Conjunctions: Assume that $\vdash \forall(\phi \leftrightarrow \phi^*)$. Then consider some derivation D in QD of $\forall(\phi \leftrightarrow \phi^*)$. The basic idea is to extend this derivation to a derivation of $\forall((\phi \wedge \psi) \leftrightarrow (\phi^* \wedge \psi))$ by first stripping away the initial quantifiers, then manipulating the truth functional connectives, and, finally, restoring the quantifiers. In detail, the new extended derivation should go (to save space when numbering lines, let $q = n + m$):

	$\vdots$	
$n.$	$\forall(\phi \leftrightarrow \phi^*)$	last line of D
$n+1.$	$\forall[(\phi \leftrightarrow \phi^*)c_1/x_1]$	$\forall$-*Elim*, n
	$\vdots$	
$n+m.$	$(\phi \leftrightarrow \phi^*)c_1 ... c_m/x_1 ... x_m$	$\forall$-*Elim*, $n+m-1$
$q+1.$	$\boxed{(\phi \wedge \psi)c_1 ... c_m/x_1 ... x_m}$	*Assume*

$q+2.$	$\phi c_1 \dots c_m/x_1 \dots x_m$	$\wedge$-*Elim*, $q+1$
$q+3.$	$\phi^* c_1 \dots c_m/x_1 \dots x_m$	$\leftrightarrow$-*Elim*, q, $q+2$
$q+4.$	$\psi c_1 \dots c_m/x_1 \dots x_m$	$\wedge$-*Elim*, $q+1$
$q+5.$	$(\phi^* \wedge \psi) c_1 \dots c_m/x_1 \dots x_m$	$\wedge$-*Intro*, $q+3$, $q+4$
$q+6.$	$(\phi \wedge \psi) c_1 \dots c_m/x_1 \dots x_m \rightarrow$	
	$(\phi^* \wedge \psi) c_1 \dots c_m/x_1 \dots x_m$	$\rightarrow$-*Intro*, $q+1$–$q+5$
$q+7.$	$(\phi^* \wedge \psi) c_1 \dots c_m/x_1 \dots x_m$	*Assume*
$q+8.$	$\phi^* c_1 \dots c_m/x_1 \dots x_m$	$\wedge$-*Elim*, $q+7$
$q+9.$	$\psi c_1 \dots c_m/x_1 \dots x_m$	$\wedge$-*Elim*, $q+7$
$q+10.$	$\phi c_1 \dots c_m/x_1 \dots x_m$	$\leftrightarrow$-*Elim*, q, $q+8$
$q+11.$	$(\phi \wedge \psi) c_1 \dots c_m/x_1 \dots x_m$	$\wedge$-*Intro*, $q+9$, $q+10$
$q+12.$	$(\phi^* \wedge \psi) c_1 \dots c_m/x_1 \dots x_m \rightarrow$	
	$(\phi \wedge \psi) c_1 \dots c_m/x_1 \dots x_m$	$\rightarrow$-*Intro*, $q+7$–$q+11$
$q+13.$	$[(\phi \wedge \psi) \leftrightarrow$	
	$(\phi^* \wedge \psi)] c_1 \dots c_m/x_1 \dots x_m$	$\leftrightarrow$-*Intro*, $q+6$, $q+12$
$q+14.$	$\forall[(\phi \wedge \psi) \leftrightarrow$	
	$(\phi^* \wedge \psi)] c_1 \dots c_{m-1}/x_1 \dots x_{m-1}$	$\forall$-*Intro*, $q+13$
	$\vdots$	
$n+2m.$	$\forall((\phi \wedge \psi) \leftrightarrow (\phi^* \wedge \psi))$	$\forall$-*Intro*, $n+2m-13$

It is important to note that all the constants introduced on lines $n+1$ through $n+m$ need to be new constants that do not appear in any previous lines. If not, then there's no guarantee that we be able to do $\forall$-*Intro* on the end lines. ■

Proof of Thm. 7.6: Just as with the proof of the Restricted Replacement Theorem for SD (Thm. 6.13, on page 156), the proof for the Replacement Theorem for QD is a recursive proof. But, since the definition of provably equivalent is different (we've extended it to formulas of QL) we can't simply extend the proof of theorem 6.13 by adding new cases to the inheritance step for the quantifiers.

Assume that ϕ and ϕ^* are provably equivalent formulas of QL (assume that $\vdash \forall(\phi \leftrightarrow \phi^*)$), that ϕ is a subformula of θ, and that θ^* is the result of replacing ϕ with ϕ^* in θ.

Base Step: Similar to the base step in the proof of theorem 6.13, in the base case θ is atomic and so has no subformula other than itself. So, if ϕ is a subformula of θ, then $\phi = \theta$. Hence $\theta^* = \phi^*$. Since ϕ and ϕ^* are provably equivalent, it follows immediately that θ and θ^* are provably equivalent.

Inheritance Step:

Recursive Assumption: Assume that the theorem holds for formulas ψ, ψ_1, ..., ψ_k; that is, assume that if ψ^* is the result of replacing ϕ with ϕ^*, then $\vdash \forall(\psi \leftrightarrow \psi^*)$, and similarly for the others.

Negation: Assume that $\theta = \sim\psi$. Either $\phi = \theta$, in which case it trivially follows that $\vdash \forall(\theta \leftrightarrow \theta^*)$, or ϕ is a subformula of ψ (and hence $\theta^* = \sim\psi^*$). By the recursive assumption, $\vdash \forall(\psi \leftrightarrow \psi^*)$. It follows by the One-step Replacement Lemma (Thm. 7.9) that $\vdash \forall(\sim\psi \leftrightarrow \sim\psi^*)$.

Conjunction: Assume that $\theta = \psi_1 \wedge ... \wedge \psi_k$. Either $\phi = \theta$, in which case it trivially follows that $\vdash \forall(\theta \leftrightarrow \theta^*)$, or ϕ is a subformula of one of the conjuncts ψ_i (and hence $\theta^* = \psi_1 \wedge ... \wedge \psi_i^* \wedge ... \wedge \psi_k$). By the recursive assumption, $\vdash \forall(\psi_i \leftrightarrow \psi_i^*)$. It follows by the One-step Replacement Lemma (Thm. 7.9) that $\vdash ((\psi_1 \wedge ... \wedge \psi_i \wedge ... \wedge \psi_k) \leftrightarrow (\psi_1 \wedge ... \wedge \psi_i^* \wedge ... \wedge \psi_k))$.

Disjunction: This case is left to the reader.

Conditional This case is also left to the reader.

Biconditional: This case is also left to the reader.

Universal: Assume that $\theta = \forall\beta\psi$. Either $\phi = \theta$, in which case it trivially follows that $\vdash \forall(\theta \leftrightarrow \theta^*)$, or ϕ is a subformula of ψ (and hence $\theta^* = \forall\beta\psi^*$). By the recursive assumption, $\vdash \forall(\psi \leftrightarrow \psi^*)$. To derive $\forall(\forall\beta\psi \leftrightarrow \forall\beta\psi^*)$, start with the derivation of $\forall(\psi \leftrightarrow \psi^*)$. Extend it by adding as many steps of $\forall$-*Elim* as needed to get to the sentence $(\psi \leftrightarrow \psi^*)\mathrm{c}_1 ... \mathrm{c}_\mathrm{m}/x_1 ... x_m$. Then using $\forall$-*Intro* first on β, then on the others we can get $\forall\forall\beta(\psi \leftrightarrow \psi^*)$ on the line. Hence $\vdash \forall\forall\beta(\psi \leftrightarrow \psi^*)$, so by the One-step Replacement Lemma (Thm. 7.9) we get that $\vdash \forall(\forall\beta\psi \leftrightarrow \forall\beta\psi^*)$.

Existential: This case is exactly the same, except that a different result from the One-step Replacement Lemma is used.

Closure Step: Since the inheritance step covers all the ways to generate QL formulas, we've shown that the theorem holds for all QL formulas θ.

■

Proof of Thm. 7.5, Part (ii): Since any two formulas ϕ and ϕ^* got by substituting QL formulas into the may-add and given schemas of the exchange shortcut rules from QD$^+$ (tables 6.40 and 7.19) are provably equivalent, it follows from the Replacement Theorem for QD (Thm. 7.6 on page 172) that if θ^* is a sentence sanctioned by an exchange rule applied to some sentence θ, then θ and θ^* are provably equivalent. That is, $\vdash \forall(\theta \leftrightarrow \theta^*)$. Since θ and θ^* are sentences, the universal closure of their biconditional $\theta \leftrightarrow \theta^*$ is just the biconditional itself, so $\vdash (\theta \leftrightarrow \theta^*)$. But since $\vdash (\theta \leftrightarrow \theta^*)$, it should be clear that $\theta \vdash \theta^*$. Thus, any application of an exchange rule from QD$^+$ is derivable using the basic rules of QD alone. ■

7.3 Exercises

7.3.1 QD Shortcut Rules

Prove that any two formulas got by substituting QL formulas into the may-add and given schemas of the exchange shortcut rules for QD are provably equivalent (Def. 7.8, on page 172). That is, show that the following hold for all QL formulas $\phi, \phi_1, \phi_2, \theta, \psi$ by writing the appropriate derivation schemas. Note that all but (7) and (8) deal with exchange shortcut rules from SD$^+$. For these virtually all the work has been done in exercise 6.4.2; all you need to do is show how to put the derivation schemas done there together and how to remove and put back on the quantifiers needed to make the universal closure.

1. $\vdash \forall[\sim(\phi_1 \wedge \phi_2) \leftrightarrow (\sim\phi_1 \vee \sim\phi_2)]$
2. $\vdash \forall[\sim(\phi_1 \vee \phi_2) \leftrightarrow (\sim\phi_1 \wedge \sim\phi_2)]$
3. $\vdash \forall[\sim\sim\phi \leftrightarrow \phi]$
4. $\vdash \forall[(\phi \rightarrow \theta) \leftrightarrow (\sim\phi \vee \theta)]$
5. $\vdash \forall[(\phi \rightarrow \theta) \leftrightarrow (\sim\theta \rightarrow \sim\phi)]$
6. $\vdash \forall[\sim(\phi \rightarrow \theta) \leftrightarrow (\phi \wedge \sim\theta)]$
7. $\vdash \forall[\sim\forall\beta\phi \leftrightarrow \exists\beta\sim\phi]$
8. $\vdash \forall[\sim\exists\beta\phi \leftrightarrow \forall\beta\sim\phi]$
9. $\vdash \forall[(\theta \wedge (\phi_1 \vee \phi_2)) \leftrightarrow ((\theta \wedge \phi_1) \vee (\theta \wedge \phi_2))]$
10. $\vdash \forall[((\phi_1 \vee \phi_2) \wedge \theta) \leftrightarrow ((\phi_1 \wedge \theta) \vee (\phi_2 \wedge \theta))]$
11. $\vdash \forall[(\theta \vee (\phi_1 \wedge \phi_2)) \leftrightarrow ((\theta \vee \phi_1) \wedge (\theta \vee \phi_2))]$
12. $\vdash \forall[((\phi_1 \wedge \phi_2) \vee \theta) \leftrightarrow ((\phi_1 \vee \theta) \wedge (\phi_2 \vee \theta))]$
13. $\vdash \forall[(\theta \leftrightarrow \psi) \leftrightarrow ((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi))]$

7.3.2 QD Practice Problems

Write derivations for each of the following using only the rules specified by the instructor. It is probably a good idea to do the problems in order, as the earlier ones tend to be easier than the later ones.

1. $\vdash \forall x\forall yQxy \rightarrow \forall zQzz$
2. $\vdash \forall x\forall yQxy \rightarrow \forall x\forall yQyx$
3. $\vdash \forall x(Qx \wedge Gx) \rightarrow (\forall xQx \wedge \forall xGx)$
4. $\vdash (\forall xQx \wedge \forall xGx) \rightarrow \forall x(Qx \wedge Gx)$
5. $\vdash (\forall xQx \vee \forall xGx) \rightarrow \forall x(Qx \vee Gx)$
6. $\vdash \forall x(Qx \rightarrow Gx) \rightarrow (\forall xQx \rightarrow \forall xGx)$
7. $\vdash \forall x(P \wedge Qx) \rightarrow (P \wedge \forall xQx)$
8. $\vdash (P \wedge \forall xQx) \rightarrow \forall x(P \wedge Qx)$
9. $\vdash \forall x(P \vee Qx) \rightarrow (P \vee \forall xQx)$
10. $\vdash (P \vee \forall xQx) \rightarrow \forall x(P \vee Qx)$
11. $\vdash \forall x(P \rightarrow Qx) \rightarrow (P \rightarrow \forall xQx)$
12. $\vdash (P \rightarrow \forall xQx) \rightarrow \forall x(P \rightarrow Qx)$

13. $\vdash \exists x \forall y Qxy \rightarrow \forall y \exists x Qxy$
14. $\vdash \forall x(Qx \rightarrow Gx) \rightarrow (\exists x Qx \rightarrow \exists x Gx)$
15. $\vdash \exists x(Qx \wedge Gx) \rightarrow (\exists x Qx \wedge \exists x Gx)$
16. $\vdash (\exists x Qx \vee \exists x Gx) \rightarrow \exists x(Qx \vee Gx)$
17. $\vdash \exists x(P \wedge Qx) \rightarrow (P \wedge \exists x Qx)$
18. $\vdash (P \wedge \exists x Qx) \rightarrow \exists x(P \wedge Qx)$
19. $\vdash \exists x(P \vee Qx) \rightarrow (P \vee \exists x Qx)$
20. $\vdash (P \vee \exists x Qx) \rightarrow \exists x(P \vee Qx)$
21. $\vdash \exists x(P \rightarrow Qx) \rightarrow (P \rightarrow \exists x Qx)$
22. $\vdash (P \rightarrow \exists x Qx) \rightarrow \exists x(P \rightarrow Qx)$
23. $\vdash \forall x(Qx \rightarrow P) \rightarrow (\exists x Qx \rightarrow P)$
24. $\vdash (\exists x Qx \rightarrow P) \rightarrow \forall x(Qx \rightarrow P)$
25. $\vdash \forall x \exists y(Qx \wedge Gy) \rightarrow (\forall x Qx \wedge \exists y Gy)$
26. $\vdash (\forall x Qx \wedge \exists y Gy) \rightarrow \forall x \exists y(Qx \wedge Gy)$
27. $\vdash \forall x \exists y(Qx \wedge Gy) \rightarrow \exists y \forall x(Qx \wedge Gy)$
28. $\vdash \forall x \exists y(Qx \vee Gy) \rightarrow (\forall x Qx \vee \exists y Gy)$
29. $\vdash (\exists y Gy \vee \forall x Qx) \rightarrow \forall x \exists y(Qx \vee Gy)$
30. $\vdash \exists y \forall x(Qx \vee Gy) \rightarrow \forall x \exists y(Qx \vee Gy)$
31. $\vdash \exists y \forall x(Qx \rightarrow Gy) \rightarrow \forall x \exists y(Qx \rightarrow Gy)$
32. $\vdash \forall x \exists y(Qx \vee Gy) \rightarrow \exists y \forall x(Qx \vee Gy)$
33. $\vdash (\exists y Qy \rightarrow \exists x Gx) \rightarrow \exists y \forall x(Qx \rightarrow Gy)$
34. $\vdash \exists y \forall x(Qx \rightarrow Gy) \rightarrow (\exists x Qx \rightarrow \exists x Gx)$
35. $\vdash \forall x \exists y(Qx \rightarrow Gy) \rightarrow \exists y \forall x(Qx \rightarrow Gy)$
36. $\vdash \exists x \exists y(Qx \wedge \sim Qy) \rightarrow (\exists x Qx \wedge \exists x \sim Qx)$
37. $\vdash \exists x \forall y(Qx \rightarrow Gy) \rightarrow (\forall x Qx \rightarrow \forall x Gx)$
38. $\vdash (\exists x Qx \wedge \exists x \sim Qx) \rightarrow \exists x \exists y(Qx \wedge \sim Qy)$
39. $\vdash (\forall x Qx \rightarrow \forall x Gx) \rightarrow \exists x \forall y(Qx \rightarrow Gy)$
40. $\vdash (\sim \exists x Qx \vee \forall x Qx) \rightarrow \forall x \forall y(Qx \rightarrow Qy)$

Chapter 8

Soundness and Completeness

8.1 Introduction

In this chapter we establish various connections between the derivation rules of the systems defined in the last two chapters with the semantic notions of truth, logical truth, and entailment. We carefully defined the rules of SD and QD so that they are *truth-preserving*.

Definition 8.1. A rule is *truth-preserving* iff the sentence or sentences to which the rule is applied entail any sentence which the rule sanctions you to write as the next step.

To prove this we start with the following theorem:

Theorem 8.2. Every application of every basic rule of SD is truth-preserving.

Proof: It can be shown that: for any basic rule R of SD, if some substitution of SL sentences into the given schema of R results in SL sentences $\psi_1, \dots, \psi_n$ and that same substitution into the may-add schema of R results in the SL sentence ϕ, then $\psi_1, \dots, \psi_n \models \phi$. Call this the truth-preservation lemma.[1] Now consider some arbitrary application of some basic rule R of SD. Say that in this application R is applied to sentences $\theta_1, \dots, \theta_m$ and permits, or sanctions, you to write down δ. By definition 6.2 (on page 130), there's some substitution of SL sentences that, for the given schemas of R, results in $\theta_1, \dots, \theta_m$ and, for the may-add schema, results in δ. By the truth preservation lemma, $\theta_1, \dots, \theta_m \models \delta$. By definition 8.1 (on the current page), this application is truth-preserving. ■

> This proof doesn't cover $\rightarrow$-*Intro* or *Assume*. These rules require special treatment. The former is the only rule that eliminates an assumption, and the latter is the only rule that adds an assumption, so they each have a unique role.

Recall from section 6.1 (on page 123) that we want to use derivations as a way of showing that a sentence is a logical truth, or of showing that some set of sentences entails some other sentence. Specifically, we want to use derivations in SD and QD to show that sentences of SL and QL are TFT and QT, or to show entailments between

[1] See exercise 2.8.8, on page 55.

sentences of SL or between sentences of QL. But derivations can only fill this role if the derivation system in question is sound. Let L be some formal language for which we have defined some kind of models.

Definition 8.3. A derivation system D for L is *sound* iff for every set Δ of sentences of L and every sentence ϕ of L, if $\Delta \vdash \phi$, then $\Delta \models \phi$.

8.2 Soundness

8.2.1 Soundness of SD

We prove the soundness of SD by way of a lemma.

Theorem 8.4. SD Soundness Theorem: SD is sound; i.e., for every set Δ of sentences of SL and every sentence ϕ of SL, if $\Delta \vdash \phi$ in SD, then $\Delta \models \phi$.

Theorem 8.5. Soundness Lemma: For any sequence of derivation lines that is a derivation, the sentence ϕ on the last line is entailed by the set Δ of sentences that are on unboxed lines and are sanctioned by *Assume*.

Since the definition of a derivation is a recursive definition[2], the most natural way to prove theorem 8.5 is through a recursive proof. The recursive proof uses three easily proved lemmas (proofs are left to the reader; the first lemma, on monotonicity, is also used to prove thm. 8.4). Two are facts about entailment and one is about derivation.

Theorem 8.6. Monotonicity of Entailment: For all SL sentences $\phi_1, \dots, \phi_n, \theta, \psi$:

$$\text{If } \phi_1, \phi_2, \dots, \phi_n \models \psi, \text{ then } \phi_1, \phi_2, \dots, \phi_n, \theta \models \psi$$

Theorem 8.7. Transitivity of Entailment: For all SL sentences $\phi_1, \dots, \phi_n$, θ, and $\psi_1, \dots, \psi_k$:

$$\begin{aligned} \text{If } \phi_1, \phi_2, \dots, \phi_n &\models \psi_1 \text{ and} \\ \phi_1, \phi_2, \dots, \phi_n &\models \psi_2 \text{ and} \\ &\vdots \\ \phi_1, \phi_2, \dots, \phi_n &\models \psi_k \text{ and} \\ \psi_1, \psi_2, \dots, \psi_k &\models \theta \quad \text{then:} \\ &\phi_1, \phi_2, \dots, \phi_n \models \theta \end{aligned}$$

Theorem 8.8. Non-decreasing Assumption Principle (NDAP): If Δ_1 is the set of assumptions of an unboxed line and Δ_2 is the set of assumptions of a later unboxed line, then Δ_1 is a subset of Δ_2, i.e., $\Delta_1 \subseteq \Delta_2$.

Proof of Thm. 8.5, Soundness Lemma:

[2] See definition 6.3, on page 130.

Base Step: Consider a single-line derivation D of the sentence ϕ sanctioned by the rule *Assume*. Since the sentence on the last line is ϕ, sanctioned by *Assume*, and $\phi \models \phi$, it follows that the set of all unboxed sentences sanctioned by *Assume* entails ϕ.

Inheritance Step: Let D be an arbitrary derivation consisting of k lines, where $k > 1$. Let Δ_i be the set of unboxed assumptions occurring in a derivation D up to (and including) line number i. Consider the case in which a new line with sentence ϕ is added to derivation D, sanctioned by rule R.

> The resulting derivation has $k+1$ lines. We want to show, for each rule R of SD, that $\Delta_{k+1} \models \phi$. Each rule is considered separately in the following.

Recursive Assumption: Assume for each line i of derivation D, where $i \leq k$ and ϕ_i is the sentence on line i, that $\Delta_i \models \phi_i$.

***Assume*:** Let line $k+1$ of D be sanctioned by *Assume*. Note that the set Δ_{k+1} of unboxed assumptions are those in Δ_k plus ϕ. Clearly $\phi \models \phi$. Then $\Delta, \phi \models \phi$ follows by monotonicity.

***Repetition*:** Let line $k+1$ of D be sanctioned by *Repetition*. Since ϕ is sanctioned at line $k+1$, then it must already occur on some line i, where $i < k+1$. By the recursive assumption $\Delta_i \models \phi$. By NDAP, $\Delta_i \subseteq \Delta_{k+1}$. So by monotonicity, $\Delta_{k+1} \models \phi$.

$\vee$-*Intro*: Let line $k+1$ of D be sanctioned by $\vee$-*Intro*. Then ϕ must be a disjunction with some disjunct, θ, that is already present on some line i, where $i < k+1$. Δ_i is the set of unboxed assumptions at that line, and by NDAP $\Delta_i \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \theta$. So by monotonicity, $\Delta_{k+1} \models \theta$. Since $\vee$-*Intro* is truth preserving, $\theta \models \phi$. So by transitivity of entailment, $\Delta_{k+1} \models \phi$.

$\wedge$-*Elim*: Let line $k+1$ of D be sanctioned by $\wedge$-*Elim*. Then there's some earlier line i with the sentence $\theta_1 \wedge ... \wedge \theta_m$ and ϕ is one of the conjuncts. As before, by NDAP $\Delta_i \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \theta_1 \wedge ... \wedge \theta_m$. So by monotonicity, $\Delta_{k+1} \models \theta_1 \wedge ... \wedge \theta_m$. And ϕ is one of the conjuncts of $\theta_1 \wedge ... \wedge \theta_m$, so it follows that $\theta_1 \wedge ... \wedge \theta_m \models \phi$. So by transitivity, $\Delta_{k+1} \models \phi$.

$\sim$-*Elim*: Let line $k+1$ of D be sanctioned by $\sim$-*Elim*. Then there's some earlier line i with the sentence $\sim\phi \rightarrow (\psi \wedge \sim\psi)$. As before, by NDAP $\Delta_i \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \sim\phi \rightarrow (\psi \wedge \sim\psi)$. So by monotonicity, $\Delta_{k+1} \models \sim\phi \rightarrow (\psi \wedge \sim\psi)$. Since the RHS of $\sim\phi \rightarrow (\psi \wedge \sim\psi)$ is false in all models (it's TFF), the conditional is true in a model $\mathfrak{m}$ only if the LHS is false in $\mathfrak{m}$. So if the conditional is true in a model $\mathfrak{m}$, ϕ is true in $\mathfrak{m}$. In other words, $\sim\phi \rightarrow (\psi \wedge \sim\psi) \models \phi$. So by transitivity, $\Delta_{k+1} \models \phi$.

$\sim$-*Intro*: This case is similar to the previous one and is left as an exercise for the reader.

$\rightarrow$-*Elim*: Let line $k+1$ of D be sanctioned by $\rightarrow$-*Elim*. Then there are two earlier lines i and j, and (say) line i has a sentence $\theta \rightarrow \phi$ and line j has sentence θ. By NDAP we have that $\Delta_i \subseteq \Delta_{k+1}$ and $\Delta_j \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \theta \rightarrow \phi$ and $\Delta_j \models \theta$. By monotonicity, $\Delta_{k+1} \models \theta \rightarrow \phi$ and $\Delta_{k+1} \models \theta$. Since the rule is truth preserving, $\theta, \theta \rightarrow \phi \models \phi$. So by transitivity, $\Delta_{k+1} \models \phi$.

$\leftrightarrow$-*Elim*: The argument for each of the two versions of $\leftrightarrow$-*Elim* is the same as that for $\rightarrow$-*Elim*.

$\leftrightarrow$-*Intro*: Let line $k+1$ of D with sentence $\phi = \phi \leftrightarrow \theta$ be sanctioned by $\leftrightarrow$-*Intro*. Then there are two earlier lines i and j, and (say) line i has a sentence $\theta \rightarrow \phi$ and line j has sentence $\phi \rightarrow \theta$. By NDAP we have that $\Delta_i \subseteq \Delta_{k+1}$ and $\Delta_j \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \theta \rightarrow \phi$ and $\Delta_j \models \phi \rightarrow \theta$. By monotonicity, $\Delta_{k+1} \models \theta \rightarrow \phi$ and $\Delta_{k+1} \models \phi \rightarrow \theta$. Since the rule is truth preserving, $\phi \rightarrow \theta, \theta \rightarrow \phi \models \phi \leftrightarrow \theta$. So by transitivity, $\Delta_{k+1} \models \phi$.

$\wedge$-*Intro*: Let line $k+1$ of D with sentence $\phi = \phi_1 \wedge ... \wedge \phi_m$ be sanctioned by $\wedge$-*Intro*. Then there are m earlier lines numbered $i_1, ..., i_m$ with, respectively, sentences $\phi_1, ..., \phi_m$. By NDAP we have that $\Delta_{i_1} \subseteq \Delta_{k+1}, ..., \Delta_{i_m} \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_{i_1} \models \phi_1, ..., \Delta_{i_m} \models \phi_m$. So by monotonicity, $\Delta_{k+1} \models \phi_1, ..., \Delta_{k+1} \models \phi_m$. Observe that $\phi_1, ..., \phi_m \models \phi \wedge ... \wedge \phi_m$. So by transitivity, $\Delta_{k+1} \models \phi$.

$\vee$-*Elim*: Let line $k+1$ of D be sanctioned by $\vee$-*Elim*. Then there are $m+1$ earlier lines numbered $i_1, ..., i_m, i_{m+1}$ with, respectively, sentences $\theta_1 \rightarrow \phi$, ..., $\theta_m \rightarrow \phi$, and $\theta_1 \vee ... \vee \theta_m$. By NDAP we have that $\Delta_{i_1} \subseteq \Delta_{k+1}, ..., \Delta_{i_m} \subseteq \Delta_{k+1}$ and $\Delta_{i_{m+1}} \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_{i_1} \models \theta_1 \rightarrow \phi$, ..., $\Delta_{i_m} \models \theta_m \rightarrow \phi$ and $\Delta_{i_{m+1}} \models \theta_1 \vee ... \vee \theta_m$. By monotonicity, $\Delta_{k+1} \models \theta_1 \rightarrow \phi$, ..., $\Delta_{k+1} \models \theta_m \rightarrow \phi$ and $\Delta_{k+1} \models \theta_1 \vee ... \vee \theta_m$. Observe that $\theta_1 \rightarrow \phi, ..., \theta_m \rightarrow \phi, \theta_1 \vee ... \vee \theta_m \models \phi$. So by transitivity, $\Delta_{k+1} \models \phi$.

$\rightarrow$-*Intro*: The rule $\rightarrow$-*Intro* changes the number of unboxed assumptions. If line $k+1$ sanctioned by $\rightarrow$-*Intro* has sentence $\phi \rightarrow \theta$ and unboxed assumptions Δ_{k+1}, then there must be an assumption line (now in a box) that starts with ϕ and Δ_{k+1} as its other assumptions, and we have a line (now at the bottom of the box) with θ on it with assumptions ϕ, Δ_{k+1}. By the recursive assumption we have that $\phi, \Delta_{k+1} \models \theta$. Consider any model $\mathfrak{m}$ that makes Δ_{k+1} true; if it also makes ϕ true, then θ is true in $\mathfrak{m}$ as well and so is $\phi \rightarrow \theta$. If $\mathfrak{m}$ makes ϕ false, then $\phi \rightarrow \theta$ is true. (Notice that this step only works because we defined the conditional to be true when the LHS is false.) So if $\mathfrak{m}$ makes Δ_{k+1} true, it makes $\phi \rightarrow \theta$ true too. So, $\Delta_{k+1} \models \phi \rightarrow \theta$.

Closure Step: We have now covered all the generating cases for derivations. By the

closure clause of the definition, we have proved soundness for all derivations.

■

Proof of Thm. 8.4, SL Soundness Theorem: Assume that Δ is a set of SL sentences. Assume $\Delta \vdash \phi$ and consider some derivation D of ϕ from Δ. Let Δ' be the set of sentences in Δ that appear as unboxed assumptions in D. By the soundness lemma (Thm. 8.5), $\Delta' \models \phi$. It follows immediately by monotonicity that $\Delta \models \phi$. ■

8.2.2 Soundness of QD

In this section we prove that QD is also sound.

Theorem 8.9. QD Soundness Theorem: QD is sound; i.e., for every set Δ of sentences of QL and every sentence ϕ of QL, if $\Delta \vdash \phi$ in SD, then $\Delta \models \phi$.

The proof given in the last section of the SL Soundness Theorem 8.4 can be carried over to the QL Soundness Theorem. That proof relied on the monotonicity of entailment and the soundness lemma (Thm. 8.5). It should be clear that entailment is also monotonic in the case of QL. Since QD is an extension of SD (it's just SD plus the rules for the quantifiers in table 7.1 on page 163), all we need to do to show that the soundness lemma holds for QD is add a case, for each new rule of QD, to the inheritance step of the proof of the soundness lemma for SD.

Proof of Thm. 8.5 for GQD:

Base Step: The base case is covered in Thm. 8.5.

Inheritance Step: Let D be an arbitrary derivation consisting of k lines, where $k > 1$. Let Δ_i be the set of unboxed assumptions occurring in a derivation D up to (and including) line number i. Consider the case in which a new line with sentence ϕ is added to derivation D, sanctioned by rule R.

> The resulting derivation has $k+1$ lines. We want to show, for each rule R of QD, that $\Delta_{k+1} \models \phi$. It was shown already that the property holds of each SD rule, so we only need to consider the quantifier introduction and elimination rules.

Recursive Assumption: Assume for each line i of derivation D, where $i \leq k$ and ϕ_i is the sentence on line i, that $\Delta_i \models \phi_i$.

$\forall$-*Elim*: Let line $k+1$ of D with sentence $\phi s/\beta$ be sanctioned by $\forall$-*Elim*. Then there's some earlier line i with the sentence $\forall\beta\phi$. Δ_i is the set of unboxed assumptions of line i, and by NDAP $\Delta_i \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \forall\beta\phi$. So by monotonicity, $\Delta_{k+1} \models \forall\beta\phi$.

Assume some model $\mathfrak{m}$ such that $\forall\beta\phi$ is true. By the def. of truth of $\forall$, $\phi t/\beta$ is true on all t-variants of $\mathfrak{m}$. Notice that $\phi t/\beta$ and $\phi s/\beta$ are exactly the same, except that the latter has s substituted for t. These sentences satisfy condition (1) of Dragnet.

Consider the t-variant that assigns to t what $\mathfrak{m}$ assigns to s. Name that t-variant $\mathfrak{m}^{\mathrm{t}}$. The models $\mathfrak{m}$ and $\mathfrak{m}^{\mathrm{t}}$ meet Dragnet condition (2). Thus, by Dragnet, $\phi t/\beta$ is true on $\mathfrak{m}^{\mathrm{t}}$ iff $\phi s/\beta$ is true on $\mathfrak{m}$. Therefore, $\phi s/\beta$ is true on $\mathfrak{m}$.

Any model such that $\forall\beta\phi$ is true also makes $\phi s/\beta$ true. Thus, $\forall\beta\phi \models \phi s/\beta$. So by transitivity, $\Delta_{k+1} \models \phi s/\beta$.

$\exists$-*Intro*: Let line $k+1$ of D with sentence $\exists\beta\phi$ be sanctioned by $\exists$-*Intro*. Then there's some earlier line i with the sentence $\phi s/\beta$. Δ_i is the set of unboxed assumptions of line i, and by NDAP $\Delta_i \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \phi s/\beta$. By monotonicity, $\Delta_{k+1} \models \phi s/\beta$.

Assume some model $\mathfrak{m}$ such that $\exists\beta\phi$ is false. By the def. of truth of $\exists$, there is no t-variant of $\mathfrak{m}$ that makes $\phi t/\beta$ true. Notice that $\phi t/\beta$ and $\phi s/\beta$ are exactly the same, except that the latter has s substituted for t. These sentences satisfy condition (1) of Dragnet.

Consider the t-variant that assigns to t what $\mathfrak{m}$ assigns to s. Name that t-variant $\mathfrak{m}^{\mathrm{t}}$. The models $\mathfrak{m}$ and $\mathfrak{m}^{\mathrm{t}}$ meet Dragnet condition (2). Thus, by Dragnet, $\phi t/\beta$ is true on $\mathfrak{m}^{\mathrm{t}}$ iff $\phi s/\beta$ is true on $\mathfrak{m}$. Therefore, $\phi s/\beta$ is false on $\mathfrak{m}$.

Any model that makes $\exists\beta\phi$ false also makes $\phi s/\beta$ false. Thus, $\phi s/\beta \models \exists\beta\phi$. So by transitivity, $\Delta_{k+1} \models \exists\beta\phi$.

$\forall$-*Intro*: Let line $k+1$ of D with sentence $\forall\beta\phi$ be sanctioned by $\forall$-*Intro*. Then there's some earlier line i with the sentence $\phi s/\beta$. Δ_i is the set of unboxed assumptions of line i, and by NDAP $\Delta_i \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \phi s/\beta$. By monotonicity, $\Delta_{k+1} \models \phi s/\beta$. However $\phi s/\beta$ does not entail $\forall\beta\phi$, so we have to do some extra work and make use of the restrictions on the rule $\forall$-*Intro*.

Let $\mathfrak{m}$ be some model that makes all of Δ_{k+1} true; and let's assume for *reductio* that $\mathfrak{m}$ that makes $\forall\beta\phi$ false. One of the restrictions for $\forall$-*Intro* is that s must not occur in $\forall\beta\phi$. Hence, by the Free Choice Theorem, $\phi s/\beta$ is false on some s-variant of $\mathfrak{m}$. Let's name that s-variant $\mathfrak{m}^{\mathrm{s}}$.

The other rule restriction for $\forall$-*Intro* is that s must not occur in Δ_{k+1}. The variant $\mathfrak{m}^{\mathrm{s}}$ differs from $\mathfrak{m}$ only on the assignment to s; otherwise, they make all the same assignments. Since s doesn't occur in Δ_{k+1} and $\mathfrak{m}$ makes all of Δ_{k+1} true, $\mathfrak{m}^{\mathrm{s}}$ also makes all of Δ_{k+1} true. The assignment $\mathfrak{m}^{\mathrm{s}}$ makes to s doesn't matter for this result.

Since $\Delta_{k+1} \models \phi s/\beta$, $\mathfrak{m}^{\mathrm{s}}$ makes $\phi s/\beta$ true. But we already showed that $\phi s/\beta$ is false on $\mathfrak{m}^{\mathrm{s}}$. From the assumption that $\mathfrak{m}$ makes $\forall\beta\phi$ is false we have proven a contradiction. So $\mathfrak{m}$ makes $\forall\beta\phi$ true.

Therefore, if $\mathfrak{m}$ is a model that makes all of Δ_{k+1} true, then $\mathfrak{m}$ makes $\forall\beta\phi$ true as well. So, $\Delta_{k+1} \models \forall\beta\phi$.

***∃-Elim*:** Let line $k+1$ of D with sentence θ be sanctioned by *∃-Elim*. Then there's some earlier line i with the sentence $\phi s/\beta \rightarrow \theta$ and an earlier line j with the sentence $\exists\beta\phi$. Δ_i is the set of unboxed assumptions of line i and Δ_j the unboxed assumptions of line j. By NDAP $\Delta_i \subseteq \Delta_{k+1}$ and $\Delta_j \subseteq \Delta_{k+1}$. By the recursive assumption, $\Delta_i \models \phi s/\beta \rightarrow \theta$ and $\Delta_j \models \exists\beta\phi$. By monotonicity, $\Delta_{k+1} \models \phi s/\beta \rightarrow \theta$ and $\Delta_{k+1} \models \exists\beta\phi$. We have make use of the restrictions on the rule *∃-Elim* to show that $\Delta_{k+1} \models \theta$.

Let $\mathfrak{m}$ be some model that makes all of Δ_{k+1} true. Since $\Delta_{k+1} \models \exists\beta\phi$, $\mathfrak{m}$ also makes $\exists\beta\phi$ true. One of the rule restrictions for *∃-Elim* is that s must not occur in $\exists\beta\phi$. Hence, by the Free Choice theorem, $\phi s/\beta$ is true on some s-variant of $\mathfrak{m}$. Name that s-variant $\mathfrak{m}^s$.

Another of the rule restrictions for *∃-Elim* is that s must not occur in Δ_{k+1}. The variant $\mathfrak{m}^s$ makes all the same assignments as $\mathfrak{m}$ except in what it assigns to s. Since $\mathfrak{m}$ makes Δ_{k+1} true and Δ_{k+1} doesn't contain s, $\mathfrak{m}^s$ also makes Δ_{k+1} true. The assignment $\mathfrak{m}^s$ makes to s doesn't make any difference.

Thus, because $\Delta_{k+1} \models \phi s/\beta \rightarrow \theta$, $\mathfrak{m}^s$ makes $\phi s/\beta \rightarrow \theta$ true. We saw earlier that $\mathfrak{m}^s$ makes $\phi s/\beta$ true, so $\mathfrak{m}^s$ makes θ true as well (def. of truth, $\rightarrow$).

According to the third rule restriction for *∃-Elim*, s must not occur in θ. Because $\mathfrak{m}^s$ makes θ true and s isn't in θ, $\mathfrak{m}$ also makes θ true. The assignment that $\mathfrak{m}^s$ makes to s is irrelevant.

So, we have shown that $\Delta_{k+1} \models \theta$.

Closure Step: We have covered all the generating cases for derivations. By the closure clause of the definition, we have proved soundness for all derivations in QD.

■

8.3 Completeness

Next we prove the completeness of SD.

Definition 8.10. A derivation system D for L is *complete* iff for every finite set Δ of sentences of L and every sentence ϕ of L, if $\Delta \models \phi$, then $\Delta \vdash \phi$.

When Δ is limited to the empty set, the result is weak completeness:

Definition 8.11. A derivation system D for L is *weakly complete* iff for every sentence ϕ of L, if $\models \phi$, then $\vdash \phi$.

The following theorem can be proved using basic results already shown. In other systems of logic, what we call completeness and weak completeness are not equivalent.

They *are* equivalent in our systems, so we do not always distinguish them. There is also a notion of 'strong' completeness, which we define shortly. SD and QD are strongly complete, but this result is not trivially equivalent to completeness. There are other systems that are complete but not strongly complete.

Theorem 8.12. SD is weakly complete iff it's complete; and likewise for QD.

Proof: ($\Leftarrow$) Assume that SD/QD is complete. Then for any finite set Δ, if $\Delta \models \phi$, then $\Delta \vdash \phi$. This includes the case when Δ is the empty set. So if $\models \phi$, then $\vdash \phi$. Hence SD/QD is weakly complete.

($\Rightarrow$) Assume that SD/QD is weakly complete: for any sentence ϕ, if $\models \phi$, then $\vdash \phi$. Assume that, for some finite set Δ of sentences and sentence ϕ, $\Delta \models \phi$. Since Δ is finite, there is a sentence δ that is a conjunction of all the sentences in Δ. We want to show that $\models \delta \rightarrow \phi$. So, assume for *indirect proof* there's some model $\mathfrak{m}$ that makes $\delta \rightarrow \phi$ false. By the definition of truth for $\rightarrow$ and $\wedge$, it follows that $\mathfrak{m}$ makes all the conjuncts of δ true and ϕ false. Then $\mathfrak{m}$ makes all the sentences in Δ true and ϕ false. But we assumed that $\Delta \models \phi$. It follows that there's no model $\mathfrak{m}$ that makes $\delta \rightarrow \phi$ false. Hence $\models \delta \rightarrow \phi$, and so by weak completeness, $\vdash \delta \rightarrow \phi$. It should be clear to the reader that if $\vdash \delta \rightarrow \phi$, then $\delta \vdash \phi$. Hence, $\delta \vdash \phi$. Finally, since $\Delta \vdash \delta$ and $\vdash$ is transitive, $\Delta \vdash \phi$. ■

Definition 8.13. A derivation system D for L is *strongly complete* iff for every set Δ of sentences of L and every sentence ϕ of L, if $\Delta \models \phi$, then $\Delta \vdash \phi$.

Strong completeness differs from (regular) completeness in that Δ is allowed to be infinite. If we limit Δ so that it must be finite (but still allow it to be empty), we get completeness. Note that both SD and QD are strongly complete, but there is no simple theorem that uses results we already have which extends weak completeness to strong completeness in the way this theorem (Thm. 8.12) extends weak completeness to (regular) completeness.

Letting Δ be infinite may seem excessive, since we have defined derivations (def. 6.3, on page 130) to have only finitely many lines. A derivation can only have finitely many assumptions. And, as we've defined the single turnstile, $\Delta \vdash \phi$ iff there's a derivation of ϕ from the sentences in Δ.

As it turns out, however, there's nothing wrong with letting Δ be infinite. Showing that there's a derivation of ϕ from the sentences in Δ doesn't require that the derivation use *all* the sentences in Δ as assumptions. In general, even when Δ is finite, any derivation of ϕ from some subset of sentences in Δ shows that $\Delta \vdash \phi$. So, if Δ is infinite and $\Delta \models \phi$, if the derivation system D is complete it follows that ϕ can be derived from some finite subset of sentences of Δ.[3]

[3]Before turning to the proofs of these theorems, some historical background might be of interest. As mentioned above (Sec. 4.1.1), quantificational languages were first developed by Frege, Peirce and Mitchell in the 1870's and 1880's. But it wasn't until David Hilbert and Wilhelm Ackermann published their hugely influential text *Grundzüge der theoretischen Logik* (Principles of Mathematical

By theorem 8.12 (on the previous page), if we show that SD is weakly complete, it follows that SD is complete. To demonstrate the weak completeness of SD we prove this intermediate result:

Theorem 8.14. The SD Weak Completeness Lemma: For any sentence ϕ of SD, either $\phi \vdash A \wedge \sim A$, or ϕ is true in some model $\mathfrak{m}$.

As an aid to prove this result, we introduce a new exchange rule for QD and then show that anything derivable with QD and this rule can be derived using QD alone. We call the rule $\leftrightarrow$-*Exchange*. (Note that every application of $\leftrightarrow$-*Exchange* is truth preserving, as the last problem in exercise 2.8.9, on page 55, extends theorem 6.11, on page 153, to it.) It's given in table 8.1. All we need to show that anything that can

Name	**Given**	**May Add**
$\leftrightarrow$-*Exchange*	$\theta \leftrightarrow \psi$	$(\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)$
	$(\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)$	$\theta \leftrightarrow \psi$

Table 8.1: $\leftrightarrow$-*Exchange*

be derived using QD and $\leftrightarrow$-*Exchange* can be derived using just QD, is to prove the following:[4]

Theorem 8.15. Any two QL formulas got by substituting other QL formulas into the may-add and given schemas of $\leftrightarrow$-*Exchange* are provably equivalent; that is, $\vdash \forall((\theta \leftrightarrow \psi) \leftrightarrow ((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)))$.

Proof: We show that $\vdash \forall((\theta \leftrightarrow \psi) \leftrightarrow ((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)))$ by giving a derivation schema, which for any two formulas θ and ψ results in the needed derivation. (Note that to save space $q = n + m$.)

Logic) in 1928 that the question of completeness was clearly formulated. While Kurt Gödel, in his 1929 doctorial dissertation (republished in 1930), is widely accepted as the first person to prove that quantificational logic is strongly complete, Church (1956: 291, fn.464) reports that the Jacques Herbrand's dissertation in 1930 had the essential material for the same proof. Further, completeness follows from results of Skolem (1967 [1928]), but since the question of completeness hadn't been clearly raised yet no one seems to have noticed. Leon Henkin (1949) later developed a method of proving completeness different from Gödels. Henkin's approach is probably the most common one used today in logic textbooks, but the proof we give here is a constructive proof closer to Gödel's original. (Ours owes much to Willard Quine's completeness proof (1982).)

[4] See section 7.2.1.

1.	$(\theta \leftrightarrow \psi)c_1 \ldots c_m/x_1 \ldots x_m$	*Assume*
2.	$(\sim\theta \leftrightarrow \sim\psi)c_1 \ldots c_m/x_1 \ldots x_m$	$\sim/\leftrightarrow$-*Intro*, 1
3.	$\sim(\theta \wedge \psi)c_1 \ldots c_m/x_1 \ldots x_m$	*Assume*
4.	$(\sim\theta \vee \sim\psi)c_1 \ldots c_m/x_1 \ldots x_m$	*DeM*, 3
5.	$\sim\theta c_1 \ldots c_m/x_1 \ldots x_m$	*Assume*
6.	$\sim\psi c_1 \ldots c_m/x_1 \ldots x_m$	$\leftrightarrow$-*Elim*, 2, 5
7.	$(\sim\theta \wedge \sim\psi)c_1 \ldots c_m/x_1 \ldots x_m$	$\wedge$-*Intro*, 5, 6
8.	$(\sim\theta \rightarrow (\sim\theta \wedge \sim\psi))c_1 \ldots c_m/x_1 \ldots x_m$	$\rightarrow$-*Intro*, 5–7
9.	$\sim\psi c_1 \ldots c_m/x_1 \ldots x_m$	*Assume*
10.	$\sim\theta c_1 \ldots c_m/x_1 \ldots x_m$	$\leftrightarrow$-*Elim*, 2, 9
11.	$(\sim\theta \wedge \sim\psi)c_1 \ldots c_m/x_1 \ldots x_m$	$\wedge$-*Intro*, 9, 10
12.	$(\sim\psi \rightarrow (\sim\theta \wedge \sim\psi))c_1 \ldots c_m/x_1 \ldots x_m$	$\rightarrow$-*Intro*, 9–11
13.	$(\sim\theta \wedge \sim\psi)c_1 \ldots c_m/x_1 \ldots x_m$	$\vee$-*Intro*, 4, 8, 12
14.	$(\sim(\theta \wedge \psi) \rightarrow (\sim\theta \wedge \sim\psi))c_1 \ldots c_m/x_1 \ldots x_m$	$\rightarrow$-*Intro*, 3–13
15.	$(\sim\sim(\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi))c_1 \ldots c_m/x_1 \ldots x_m$	$\rightarrow/\vee$-*Exch.*, 14
16.	$((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi))c_1 \ldots c_m/x_1 \ldots x_m$	$\sim\sim$-*Elim*, 15
17.	$[(\theta \leftrightarrow \psi) \rightarrow ((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi))]c_1 \ldots c_m/x_1 \ldots x_m$	$\rightarrow$-*Intro*, 1–16
18.	$((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi))c_1 \ldots c_m/x_1 \ldots x_m$	*Assume*
	$\vdots$	
n.	$(\theta \leftrightarrow \psi)c_1 \ldots c_m/x_1 \ldots x_m$	
$n+1$.	$[((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)) \rightarrow (\theta \leftrightarrow \psi)]c_1 \ldots c_m/x_1 \ldots x_m$	$\rightarrow$-*Intro*, 18–n
$n+2$.	$[(\theta \leftrightarrow \psi) \leftrightarrow ((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi))]c_1 \ldots c_m/x_1 \ldots x_m$	$\leftrightarrow$-*Intro*, 17, $n+1$
	$\vdots$	
$q+2$.	$\forall((\theta \leftrightarrow \psi) \leftrightarrow ((\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)))$	$\forall$-*Intro*, $q+1$

Note that we have left steps 18–n for the reader; this is just the derivation of the other conditional needed for $\leftrightarrow$-*Intro* on line $n+2$. Also note that the last steps, lines $n+3$

to the end, are all $\forall$-*Intro* meant to eliminate the constants $c_1, \ldots, c_m$. ■

Proof of Thm. 8.14: To prove the theorem, we shall describe an algorithm for applying the rules of SD$^+$ and $\leftrightarrow$-*Exchange* that takes a SL sentence ϕ and either halts in a derivation of $A \wedge \sim A$, or halts with a sentence in DNF for which there is some model $\mathfrak{m}$ that makes ϕ true. Since a sentence can be derived using the rules of SD$^+$ and $\leftrightarrow$-*Exchange* iff it can be derived using the basic rules of SD, this is sufficient to prove the theorem.

The algorithm begins with ϕ as an assumption on line 1. The algorithm then applies the method studied earlier in section 2.6.2 (on page 47) to produce a sentence ϕ' in DNF that's TFE to ϕ. We have to show that each step of the earlier method can be carried out in steps using the rules of SD$^+$ and $\leftrightarrow$-*Exchange*. The earlier method proceeded in three stages.

Step A:

(1) If a subsentence of ϕ has $\rightarrow$ as its main connective, i.e. if $\phi = \theta \rightarrow \psi$, replace the subsentence by $\sim\theta \vee \psi$. Repeat as necessary to obtain a sentence ϕ^* without conditionals. Each of these steps are sanctioned by $\rightarrow/\vee$-*Exchange*.

(2) If a subsentence of ϕ has $\leftrightarrow$ as its main connective, i.e. if $\phi = \theta \leftrightarrow \psi$, it is replaced with the subsentence $(\theta \wedge \psi) \vee (\sim\theta \wedge \sim\psi)$. Repeat as necessary to obtain a sentence ϕ^{**} without biconditionals. Each of these steps are sanctioned by $\leftrightarrow$-*Exchange*.

Step B: In the case where ϕ^{**} contains a subsentence whose main connective is negation and which contains other connectives, we replace that subsentence by the following steps:

(1) Replace $\sim\sim\theta$ by θ; this step is sanctioned by $\sim\sim$-*Elim*.

(2) Replace $\sim(\theta \wedge \psi)$ by $\sim\theta \vee \sim\psi$; this step is sanctioned by *DeM*.

(3) Replace $\sim(\theta \vee \psi)$ by $\sim\theta \wedge \sim\psi$; this step is sanctioned by *DeM*.

Repeat as necessary to obtain a sentence ϕ^{***} in which negations govern nothing but sentence letters.

Step C: The only thing that could prevent ϕ^{***} from being in DNF is that some conjunctions govern some disjunctions, i.e., there is a subsentence of the form $\theta \wedge (\psi_1 \vee \ldots \vee \psi_n)$, or the reverse $(\psi_1 \vee \ldots \vee \psi_n) \wedge \theta$. Those subsentences can each be replaced by the equivalent sentence $(\theta \wedge \psi_1) \vee \ldots \vee (\theta \wedge \psi_n)$ or $(\psi_1 \wedge \theta) \vee \ldots \vee (\psi_n \wedge \theta)$. These steps are sanctioned by *Distribution*.

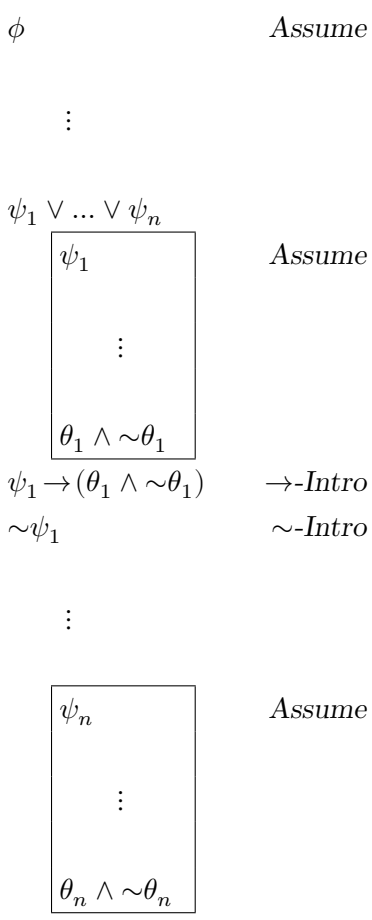
Applying the above steps A, B, and C provides us a derivation starting with ϕ as an assumption (and no other assumptions) and ending with a DNF sentence that's TFE to ϕ. There are two cases:

Case 1: Every disjunct contains a sentence letter and the negation of that sentence letter. That is, each disjunction has the form $(\psi_1 \wedge ... \wedge \psi_i \wedge ... \wedge \sim\psi_i \wedge ... \wedge \psi_n)$; for example: $(A \wedge B \wedge C \wedge \sim E \wedge \sim B \wedge \sim K)$.

Case 2: At least one disjunct contains no sentence letter such that the negation of the sentence letter is also in the disjunct.

It can be shown in case 1 that the algorithm derives a contradiction from the original sentence. First, observe that any conjunction that contains a sentence letter and its negation leads to a contradiction by repeated steps of $\wedge$-*Elim*. Thus we can derive the negation of any such conjunction using $\sim$-*Intro*. So if the last line of the derivation so far is of the form $(\psi_1 \vee ... \vee \psi_n)$ and each ψ_i contains a sentence letter and the negation of that sentence letter, then add to the derivation lines that establish the negation of each ψ_i. Thus by $n-1$ steps of *D.S.* the result is a single ψ_i by itself with only the first line as an assumption. Since this remaining conjunct has both a sentence letter and its negation it is trivial to derive a contradiction.

Schematically, the procedure looks like:



$\psi_n \rightarrow (\theta_n \wedge \sim\theta_n)$	$\rightarrow$-*Intro*
$\sim\psi_n$	$\sim$-*Intro*
$\psi_1 \vee ... \vee \psi_{n-1}$	*D.S.*
$\psi_1 \vee ... \vee \psi_{n-2}$	*D.S.*
$\psi_1 \vee ... \vee \psi_{n-3}$	*D.S.*
$\vdots$	
$\psi_1 \vee \psi_2$	*D.S.*
ψ_1	*D.S.*
$\sim\psi_1$	*Rep.*
$A \wedge \sim A$	*A.C.*

So much for case 1.

In case 2 it can be shown that there is a model that makes all sentences in the derivation true, starting with the last. To construct this model, first choose the disjunction that does not contain a sentence letter and its negation. If there is more than one, it doesn't matter which is chosen. Then construct a model $\mathfrak{m}$ by assigning `T` to each sentence letter that occurs positively (without a negation in front) and `F` to each sentence letter that occurs negatively (with a negation in front).

This model makes each element of the conjunction true and thus makes the entire conjunction true. Since the sentence containing it is a disjunction, this is sufficient to make the entire sentence true. Thus it makes the last line of the derivation true. Observe that all of the steps we used in the derivation were replacement of provably equivalence sentences; that is, they used exchange shortcut rules. Thus, we can also construct a derivation by 'turning this proof upside down', so to speak. In other words, we can construct a new derivation, with the last step of the original derivation as the initial assumption step, then use the exchange rules to work back to original ϕ.

Thus, by soundness, if the first sentence of the 'upside-down derivation' (the sentence in DNF that was at the bottom) is true in a model, then so is everything that can be derived from it, including our original sentence ϕ that is now at the end of the inverted derivation. Therefore, ϕ is true in some model. ■

Theorem 8.16. Weak SD Completeness Theorem: For all SL sentences ϕ: if $\models \phi$, then $\vdash \phi$ in SD.

Proof: We apply the method above from the SD Completeness Lemma to the negation of ϕ. This either produces a derivation of a contradiction from $\sim\phi$, in which case we can prove ϕ by adding two more steps justified by $\rightarrow$-*Intro* and $\sim$-*Elim*, or it produces a model that makes $\sim\phi$ true and that therefore makes ϕ false. So, ϕ is either false in some model or is derivable in SD. ■

Finally, as a corollary we get:

Theorem 8.17. SD Completeness Theorem: For every finite set Δ of sentences of SL and every sentence ϕ of SL, if $\Delta \models \phi$, then $\Delta \vdash \phi$ in SD.

Proof: This follows immediately from the Weak SD Completeness Theorem and theorem 8.12 (on page 186). ■

8.4 Completeness of QD

Next we prove the completeness of QD. We want to use a similar strategy for QD we did for SD, but we have to deal with the quantifiers somehow. Unfortunately, the quantifiers prevent us from proving the analogue of DNF for QL. Given an arbitrary sentence, there is no guarantee that we can reorder the quantifiers so that each has its scope contained in a single disjunct. For example: $\forall x(Hx \vee Gx)$ is not equivalent to $\forall xHx \vee \forall xGx$.

To get around problems like this, we instead show we can move all the quantifiers to the front of the sentence. This has the advantage of separating the quantifier parts of the logical structure from the SL parts.

8.4.1 Prenex Definition and Steps

Definition 8.18. A sentence ϕ of QL is in *prenex normal form* iff every quantifier is in initial position, or, in other words, the scope of all quantifiers is greater than that of any non-quantifier connective.

Theorem 8.19. Prenex Normal Form Theorem: For each sentence θ of QL, there is a provably equivalent sentence θ^* in prenex normal form; that is, θ^* is in prenex normal form and $\vdash \theta \leftrightarrow \theta^*$ in QD.

Proof: As with DNF there are a set of steps for turning sentence θ into a sentence θ^* in prenex normal form. First we give the steps, and then show that each step can be sanctioned either by *QN*, *↔-Exchange*, or an exchange rule that can be introduced. (We call these new exchange rules the *Prenex Exchange Rules*.) Because all the steps in the process are justified by exchange rules, we can either read the resulting series of steps top-down as a derivation of θ^* from θ, or bottom-up as a derivation of θ from θ^*. So, we'll have shown that $\vdash \theta \leftrightarrow \theta^*$ in the derivation system consisting of *↔-Exchange* and the Prenex Exchange Rules. But, as with all the other exchange rules anything that can be derived using the Prenex Exchange Rules can be derived in QD alone; so, this is sufficient to show that $\vdash \theta \leftrightarrow \theta^*$ in QD. First, the steps are:

(1) Replace biconditionals with disjunctions of conjunctions; i.e. replace $\phi \leftrightarrow \psi$ with $(\phi \wedge \psi) \vee (\sim\phi \wedge \sim\psi)$.

(2) Rewrite any variables that occur bound by more than one quantifier.

(3) Move the first quantifier not in prenex position one step towards the front by the following principles. Repeat this step as often as necessary. Keep in mind that you can't move forward a quantifier that binds the variable 'x' if it has within its scope a new subformula that has a free 'x'. But we have prevented that problem by eliminating potentially clashing variables in Step 2.

Replace	**by**
$((\#x)\theta \wedge \psi)$	$(\#x)(\theta \wedge \psi)$
$(\theta \wedge (\#x)\psi)$	$(\#x)(\theta \wedge \psi)$
$((\#x)\theta \vee \psi)$	$(\#x)(\theta \vee \psi)$
$(\theta \vee (\#x)\psi)$	$(\#x)(\theta \vee \psi)$
$(\theta \rightarrow (\#x)\psi)$	$(\#x)(\theta \rightarrow \psi)$
$(\exists x\theta \rightarrow \psi)$	$\forall x(\theta \rightarrow \psi)$
$(\forall x\theta \rightarrow \psi)$	$\exists x(\theta \rightarrow \psi)$
$\sim\exists x\theta$	$\forall x\sim\theta$
$\sim\forall x\theta$	$\exists x\sim\theta$

Note that $(\#x)$ is just a dummy quantifier standing for either; replacement is the same for both quantifiers. Also, we use 'x' in the chart above, but the same principles hold for quantifiers with any other variable.

After applying these steps to a sentence θ we get a sentence θ^* that is in prenex normal form.[5] We have to show that each step can be sanctioned by an exchange rule. Step (1) is straightforward, since it is sanctioned by $\leftrightarrow$-*Exchange*. But steps (2) and (3) we need new rules (although the replacements involving negations in (3) can be handled with *QN*). The most straightforward strategy is to read the needed exchange rules right off the steps. Thus, the Prenex Exchange Rules are given in the following chart.

Name	**Given**	**May Add**
α/β-*Exch*	$(\#\alpha)\phi$	$(\#\beta)\phi\beta/\alpha$
Q Shuffling	$((\#x)\theta \wedge \psi)$	$(\#x)(\theta \wedge \psi)$

Table 8.5: Prenex Exchange Short-Cut Rules for QD

Continued next Page

[5] For more discussion of Prenex Form, see Kleene 2002 [1967]: 132, Hodges 2001b: 54, 2001a: 30.

Continued from Previous Page

Name	**Given**	**May Add**
	$(\theta \wedge (\#x)\psi)$	$(\#x)(\theta \wedge \psi)$
	$((\#x)\theta \vee \psi)$	$(\#x)(\theta \vee \psi)$
	$(\theta \vee (\#x)\psi)$	$(\#x)(\theta \vee \psi)$
	$(\theta \rightarrow (\#x)\psi)$	$(\#x)(\theta \rightarrow \psi)$
	$(\exists x\theta \rightarrow \psi)$	$\forall x(\theta \rightarrow \psi)$
	$(\forall x\theta \rightarrow \psi)$	$\exists x(\theta \rightarrow \psi)$

Table 8.5: Prenex Exchange Shortcut Rules for QD

Finally we show that anything that can be derived using the Prenex Exchange Rules can be derived using the basic rules of QD alone. Recall from section 7.2.1 that all we need to do to show this is to prove the following:

Theorem 8.20. For all Prenex Exchange Rules R, any two QL formulas got by substituting other QL formulas into the may-add and given schemas of R are provably equivalent.

We leave the proof of this theorem to the reader, since as with the other exchange rules it just involves writing down the appropriate derivation schemas. ■

8.4.2 The Strategy for Proving QD Completeness

Our goal is to prove the strong completeness of QD: for any set Δ of QL sentences and QL sentence ϕ, if $\Delta \models \phi$, then $\Delta \vdash \phi$. First we prove the completeness of QD and then show how to modify the method to prove strong completeness. To prove completeness we show that for any sentence we can either (1) find a derivation of it or (2) prove that there is a model that makes it false. (This part of the strategy is more or less the same as what we did to show that SD is complete.) In other words, ϕ is either derivable or not quantificationally true, from which it immediately follows that if ϕ is quantificationally true, then it is derivable.

The method, in brief, is to negate the sentence ϕ and begin a derivation. Then we transform the negation of the sentence into prenex normal form, using the steps outlined in section 8.4.1. Next we transform the inner part of the sentence (remember the quantifiers are all up front) into DNF form, using our standard method for that (see Sec. 2.6.2, on page 47). This does not introduce any new assumptions. We then systematically take instances of the bound variables and try to derive a contradiction. If we can derive a contradiction we can then (assuming all goes well) use $\sim$-*Elim* to obtain a derivation of ϕ.

We must be very systematic since we have to be sure that if we get a contradiction we can derive it from the initial sentence ϕ; and that if we do not get a contradiction we have not overlooked anything and that we can show the existence of a model making the sentence on the first line, ϕ, true.

It is important that we know the form of the sentence we have reached and are able to prescribe a uniform systematic method. The sentence has been highly standardized; there are no biconditionals or conditionals (these have been eliminated in early steps of the transformation), negations govern only atomic sentences, and conjunctions govern only atomic sentences or their negations. These last, atomic sentence and their negations, are called *ions*. We say an ion occurs *positively* iff it's an atomic sentence without a negation, and it occurs *negatively* iff it's a negated atomic sentence. We call the quantifier free part of the original sentence the *matrix*. It is usually not a sentence since it may have free variables. We call the sentences that are obtained from the matrix by substitution in the process of constructing the derivation *matrix instances*. To put some of our jargon together, the matrix of the sentence consists of disjunctions of conjunctions of ions.

8.4.3 The Method and Completeness Lemmas

In this section we describe the method sketched above. Given a sentence θ, the Method either produces a derivation of θ or indicates a model that makes it false:

Step 0: Write $\sim\theta$ on line 1 as an assumption. Then first apply the prenex steps to put $\sim\theta$ in Prenex Normal Form (PNF). Next, apply the disjunctive normal form steps to the inner, quantifier-free part of the sentence until it's in DNF. At this point we'll have a sentence $(\sim\theta)^*$ that's in what we'll call *prenex disjunctive normal form* (PDNF).

Step 1: We continue the derivation operating on $(\sim\theta)^*$, the PDNF of the sentence we're concerned with. If this PDNF is a universal statement and contains no constants we write as the next line the instance of it we obtain by eliminating the quantifier and substituting the constant a for the previously bound variable; these steps are sanctioned by $\forall$-*Elim*. This step is only done once, whereas the next three steps generally require repeated recursive applications.

Step 2: For every universal sentence that appears thus far in the derivation, we add *all new instances* that can be formed with constants that occur earlier in the derivation; these steps are sanctioned by $\forall$-*Elim*. E.g., if $\forall x\phi$ appears on a line and the constant c appears anywhere (earlier) in the derivation, then if we have not taken an instance of ϕ with c yet (i.e., $\phi c/x$), we do so. As a practical matter, this means that it is useful in following the method to keep track somewhere of the constants used at each stage and of which constants have been used to instantiate which universal statements. Note that in this step we are taking new instances with old constants and that we are not adding any new assumptions. We may, however, be adding new existentials.

Step 3: For every existential sentence that appears in the derivation for which no instance has been added yet, add an instance using the first constant which *does not occur in any previous assumption*. Note that 'instance' is to be taken very strictly here. The fact that we instantiated $\exists x Kx$a with Kba takes care of that existential, but if we later add the sentence $\exists x Kx$b then we must add an instance of it. The rule which sanctions these steps is *Assume*. We eventually discharge these premises by $\rightarrow$-*Elim* and $\exists$-*Elim* if we get to a contradiction. It is in anticipation of this eventuality that we carefully chose a constant which does not occur in any previous assumption. Note that with this step we are adding new instances with new constants in new assumptions.

Step 4: Determine whether the conjunction of the *instances* of the matrix in the derivation thus far are contradictory. Officially, the way to do this is to take the conjunction of them all by $\wedge$-*Intro*, use *Distribution* to get the conjunction into DNF and check whether every disjunct contains a contradiction. If so, then by a process of $\vee$-*Elim* and *Any Contradiction* we can eventually produce the line $A \wedge \sim A$.

This step can be cumbersome in practice. We give some unofficial short cuts to make this more manageable soon.

Step 5:

(1) If the matrices are contradictory (see step 4) we stop.

(2) Or if the conjunction of the matrix instances is consistent and the last applications of Steps 2 and 3 produce no new sentences, we stop.

(3) Or if the conjunction of the matrix instance is consistent and the last applications of Steps 2 and 3 produced new sentences, then we return to Step 2 and reapply those steps.

There are three possible outcomes of applying this method to a sentence:

(1) The method reaches a contradiction.

(2) The method stops without a contradiction.

(3) The method generates new sentences perpetually without contradiction.

We show first that if a contradiction is reached we can construct a derivation of θ.

Theorem 8.21. Derivational Lemma: If the Method starts with $\sim\theta$ and produces a contradiction, then there is a derivation of θ.

Proof: Step 3 left us with $A \wedge \sim A$ on a line with its assumptions being those of the matrices. We want to shift those assumptions so that we end up with the contradiction from the first assumption, $\sim\theta$, alone. We know by considering our method that other assumptions entered only by Step 2, where we added instances of existentials using

new constants. We eliminate the last assumption by a $\rightarrow$-*Intro*. We know that this eliminated assumption introduced a *new* constant from an existential. Therefore we know that this constant did not appear in any earlier assumption or in the existential of which we are taking an instance. It also (obviously) does not occur in $A \wedge \sim A$. Thus we are allowed to use the rule $\exists$-*Elim* on the existential claim from which we assumed that instance.

So we derive the contradiction $A \wedge \sim A$ again, by $\exists$-*Elim*. We continue this process, repeating $A \wedge \sim A$ as often as necessary to shift the dependence back to the assumption on line 1. This gives us a derivation of $A \wedge \sim A$ from the first assumption, $\sim\theta$, only.

We then add two more lines: $\sim\theta \rightarrow (A \wedge \sim A)$, sanctioned by $\rightarrow$-*Intro*, and θ, sanctioned by $\sim$-*Elim*. Thus we have a derivation of θ from no assumptions. ■

We have shown that if we obtain a contradiction in the derivation process we can derive the original sentence that interests us.

We must now show that if we do not obtain a contradiction (whether or not the method stops), then there is a model that makes $\sim\theta$ true (and hence makes θ false).

Before giving the rigorous version of the construction of the model, we present some of the ideas in a more concrete context. If we consider a sentence such as $(K\mathrm{a} \wedge \sim G\mathrm{b}) \vee (\sim K\mathrm{b} \wedge H\mathrm{c})$ we can observe several things. First, each disjunct is satisfiable iff no ion occurs both positively and negatively in it. It is obvious that a conjunction that includes a sentence and its negation cannot be satisfied, but we can show for a conjunction of ions that that is the only way in which it can fail to be satisfiable. For example, we can make $K\mathrm{a}$ and $\sim G\mathrm{b}$ true by letting K be interpreted as the set of even numbers, G the set of numbers divisible by 10 and letting 'a' be assigned 2 and 'b' be assigned 7.

Of course several such sentences taken together produce different results. E.g., as we saw above $(K\mathrm{a} \wedge \sim K\mathrm{b} \wedge \sim G\mathrm{c}) \vee (G\mathrm{b} \wedge \sim G\mathrm{c})$ is satisfiable, as is $(K\mathrm{b} \wedge \sim K\mathrm{c} \wedge \sim G\mathrm{b}) \vee (G\mathrm{c} \wedge \sim G\mathrm{d})$, but the two together (taken as a conjunction) are not. The reason is that while each disjunct of the first sentence is self-consistent, it cannot be true simultaneously with either of the disjuncts of the second sentence. If we have a series of disjunctions then they are simultaneously satisfiable only if we can find a way of picking a disjunct from each one in such a way that all the chosen disjuncts can be true together.

This is relevant to the task at hand because we know that all of the non-quantified sentences in our derivation are in DNF and are thus disjunctions of conjunctions of ions. We are calling the quantifier free part of the original sentence the matrix. It is usually not a sentence since it may have free variables. The sentences that are obtained from the matrix by substitution in the process of constructing the derivation are the matrix instances. We use the notation $M_{i,j}$ for the disjuncts of the matrix instances, specifically the disjuncts of the first matrix instance are $M_{1,1}, M_{1,2}, \ldots, M_{1,m}$. Thus the first matrix instance is $M_{1,1} \vee M_{1,2} \vee \ldots \vee M_{1,m}$. The matrix instances that appear in the derivation can be listed in an array:

$$M_{1,1} \vee M_{1,2} \vee ... \vee M_{1,m}$$
$$M_{2,1} \vee M_{2,2} \vee ... \vee M_{2,m}$$

$$\vdots$$

$$M_{n,1} \vee M_{n,2} \vee ... \vee M_{n,m}$$

Note that if the the method never stops, then this array is infinitely long.

The matrices are jointly consistent iff there is a way of picking an $M_{i,j}$ from each matrix instance so that the conjunction of those $M_{i,j}$ contains no atomic sentence and its negation. In one direction this is easy to see: if there is no way of choosing a disjunct from each matrix instance that does not end up with an atomic sentence and its negation among the chosen sentences then the set of instances is inconsistent.

To show that all instances are satisfiable when such a selection can be made without choosing a sentence and its negation takes some proving. In order to do this we need to define the *master matrix list* M. We first choose (if there is more than one) a set of disjuncts $M_{i,j}$ (including one from each matrix instance M_i) that does not contain any atomic sentence and its negation. This is a set of conjunctions of atomic sentences and negations of atomic sentences. Our master matrix list M simply consists of all these atomic sentences and negated atomic sentences. Note that since the $M_{i,j}$ selections must be consistent no atomic sentence that appears unnegated also appears negated.

Definition 8.22. Given a master matrix list M, the *matrix model of* M is the model $\mathfrak{m}_M$ such that:

(1) The universe of $\mathfrak{m}_M$ contains one natural number for each constant that appears in M, and $\mathfrak{m}_M(\text{a}) = 1$, $\mathfrak{m}_M(\text{b}) = 2$, $\mathfrak{m}_M(\text{c}) = 3$, $\mathfrak{m}_M(\text{d}) = 4$, and so on; $\mathfrak{m}_M(t) = 1$ for any constant t that doesn't appear in M.

(2) For each m-place predicate P, $\mathfrak{m}_M(P)$ is the set of m-tuples of natural numbers $\langle n_1, ..., n_m\rangle$ such that $\mathfrak{m}_M(t_1) = n_1, ..., \mathfrak{m}_M(t_m) = n_m$ and $Pt_1 ... t_m$ appears on the list M.

(3) Assignments are only made if justified by these principles.

A bit more informally, we list the constants that occur on the master matrix list. $\mathfrak{m}_M$ has a universe that contains as many natural numbers as constants used. We assign to each constant that occurs on the list the natural number that indicates its place in the order, i.e. 1 to a, 2 to b, and so on. Note that this produces a *census*. Each 1-place predicate is assigned the set of numbers associated with the constants such that an instance of the predicate followed by that constant appears on the master matrix list M. Each 2-place predicate is assigned the set of pairs of numbers associated with constants such that an instance of the predicate followed by that pair of constants

appears on the master matrix list M. E.g., if Kab, Kbc, and Kde appear on M, then $\mathfrak{m}_M(K)$ is assigned $\{\langle 1,2\rangle, \langle 2,3\rangle, \langle 4,5\rangle\}$. Assignments are made in a similar fashion for n-placed predicates for $n > 2$.[6]

Theorem 8.23. The Method Lemma 1: The matrix model $\mathfrak{m}_M$ makes true all sentences on the master matrix list M.

Proof: By construction, if an atomic sentence appears on the list we decided to put the relevant pair, triple, or whatever, of numbers in the set assigned to the predicate letter. For each negated atomic sentence on the list we know that we would not put the relevant pair, triple, or whatever, in the model of the predicate letter unless the atomic sentence which is being negated also appeared. But that never happened because M is consistent by hypothesis. ■

Theorem 8.24. The Method Lemma 2: All matrix instances in the derivation are true in the matrix model $\mathfrak{m}_M$.

Proof: By Lemma 1 (Thm. 8.23), all sentences on the master matrix list M are true, and we included all the conjuncts of at least one disjunct $M_{i,j}$ from each matrix instance in forming the master list. ■

Theorem 8.25. The Method Lemma 3: All quantified sentences in the derivation are true in the matrix model $\mathfrak{m}_M$.

Informally, every universal has been instantiated with all the relevant constants, and every existential by at least one. Since $\mathfrak{m}_M$ is a census, it follows that all the sentences in the derivation are true.

More rigorously...

Proof: We prove this lemma using a recursive proof on the number of quantifiers in each sentence.

Base Case: The base case is the case of the sentences with $k = 0$ quantifiers. But these sentences are just the matrix instances in the derivation. We already proved in The Method Lemma 2 (Thm. 8.24) that all these sentences are true the matrix model $\mathfrak{m}_M$, so the base case is complete.

Inheritance Step:

Recursive Assumption: Our recursive assumption is that all sentences in the derivation with less than k quantifiers are true in the matrix model $\mathfrak{m}_M$.

[6]Note that if the method never stops, then the master matrix list M is infinite and we won't actually be able to write down the matrix model $\mathfrak{m}_M$. But this isn't a problem, the matrix model $\mathfrak{m}_M$ still exists, even if we can't write it down.

Existential Quantifier: Say θ is a sentence appearing in the derivation of the form $\exists\alpha\phi$, where ϕ is a formula with $k-1$ quantifiers. Step 3 of the method guarantees that the sentence $\phi t/\alpha$, for some constant t, appears somewhere in the derivation. This sentence $\phi t/\alpha$ has $k-1$ quantifiers, so by the recursive assumption it is true in the matrix model $\mathfrak{m}_M$. But then the existentially quantified sentence $\exists\alpha\phi$ is true in $\mathfrak{m}_M$ as well. (To show this rigorously, consider the s-variant of $\mathfrak{m}_M$, $\mathfrak{m}^{s_1}$, that assigns the same element of the universe of $\mathfrak{m}_M$ to s as $\mathfrak{m}_M$ assigns to the constant t. Now by the Dragnet Theorem, Thm. 4.16 on page 92, since $\mathfrak{m}_M$ makes $\phi t/\alpha$ true, the sentence $\phi s/\alpha$ is true on $\mathfrak{m}^{s_1}$. It follows from this that $\exists\alpha\phi$ is true on $\mathfrak{m}_M$.)

Universal Quantifier: Say θ is a sentence appearing in the derivation of the form $\forall\alpha\phi$, where ϕ is a formula with $k-1$ quantifiers. All instances $\phi t/\alpha$ which appear in the derivation have $k-1$ quantifiers, and so by the recursive hypothesis are true in the matrix model $\mathfrak{m}_M$. If we consider any s-variant of $\mathfrak{m}_M$, we know that what it assigns to s must be a number from the universe of $\mathfrak{m}_M$; we also know from the way that we constructed the matrix model $\mathfrak{m}_M$ that a number was included in the universe of $\mathfrak{m}_M$ only if it was assigned to some constant that occurred in the derivation. Let what's assigned to s by some s-variant, $\mathfrak{m}^{s_2}$, be the number associated with the constant c. Because the universal statement $\forall\alpha\phi$ occurred in the derivation, we know that we took all instances of it, including $\phi\mathrm{c}/\alpha$. As already stated, all instances $\phi t/\alpha$ are true in $\mathfrak{m}_M$, including $\phi\mathrm{c}/\alpha$. By the Dragnet Theorem (Thm. 4.16 on page 92), since $\phi\mathrm{c}/\alpha$ is true in $\mathfrak{m}_M$ it follows that $\phi s/\alpha$ is true on $\mathfrak{m}^{s_2}$. But the exact same argument works for every s-variant of $\mathfrak{m}_M$; so $\phi s/\alpha$ is true on every s-variant of $\mathfrak{m}_M$. It follows that $\forall\alpha\phi$ is true on the matrix model $\mathfrak{m}_M$.

Closure Step: Every sentence in the derivation is true in the matrix model $\mathfrak{m}_M$, which is what was to be shown.

■

8.4.4 Proving Completeness

In this section we put together all the pieces from the last section to prove that QD is complete.

Theorem 8.26. Main Weak QD Completeness Lemma: For all sentences θ of QL, if the method is applied to $\sim\theta$ then either: (a) the method produces a derivation of θ in QD^+, or (b) there is some model $\mathfrak{m}$ that makes θ false.

Proof: If the method is applied to $\sim\theta$, then either (1) it produces a contradiction $A\wedge\sim A$, (2) the method halts without a contradiction, or (3) the method never halts (and hence never halts in a contradiction). If (1), then by the Derivational Lemma (Thm. 8.21, on page 196) there is a derivation of θ.

If either (2) or (3) is the case, then by the Method Lemma 3 (Thm. 8.25, on page 199) we know that all sentences in the derivation starting with $(\sim\theta)^*$, the prenex disjunctive normal form sentence produced in Step 0 of the method from the sentence $\sim\theta$ on line 1, are true in the matrix model $\mathfrak{m}_M$. Note that all the steps in the derivation of $(\sim\theta)^*$ from $\sim\theta$ are sanctioned by exchange rules; therefore those steps can be turned upside down to produce a derivation in QD^+ of $\sim\theta$ from $(\sim\theta)^*$. So by theorem 7.3 (on page 171) there's a derivation in QD of $\sim\theta$ from $(\sim\theta)^*$. Since QD is sound (Thm. 8.9, on page 183), it follows that $(\sim\theta)^* \models \sim\theta$. Since we know that $(\sim\theta)^*$ is true in the matrix model $\mathfrak{m}_M$, it follows that $\sim\theta$ is true in $\mathfrak{m}_M$ too. So it follows that θ is false in $\mathfrak{m}_M$. ■

Theorem 8.27. Weak QD Completeness Theorem: For all sentences θ of QL, if $\models \theta$, then $\vdash \theta$ in QD.

Proof: Assume that $\models \theta$. Then there are no models $\mathfrak{m}$ which makes θ false. Thus, if the method is applied to $\sim\theta$ it can't be that some model $\mathfrak{m}$ makes θ false. By the Main QD Weak Completeness Lemma (Thm. 8.26), it follows that when the method is applied to $\sim\theta$ it produces a derivation of θ in QD^+. Hence there is a derivation of θ in QD. ■

Theorem 8.28. QD Completeness Theorem: For all finite sets Δ of QL sentences and QL sentence ϕ, if $\Delta \models \phi$, then $\Delta \vdash \phi$.

Proof: The theorem follows immediately from the Weak QD Completeness Theorem and theorem 8.12 (on page 186). ■

A consequence of our Strong Method is that if ϕ is entailed by an infinite set of sentences Δ, it is entailed by and derivable from a finite subset of Δ. If the Strong Method does not go on forever, then we get a contradiction at a finite stage and we have only assumed a finite subset of Δ.

8.4.5 Shortcut Rules for the Method

The method discussed in section 8.4.3 becomes practically unwieldy. For example, if the matrix has three disjuncts with two sentences each, then combining two instances gives 9 disjuncts with 4 elements each, and combining three gives 27 disjuncts with 8 elements each. Thus we use some additional short cut rules to speed the process of detecting contradictions. (But note that *two* of the shortcut Rules we add here are not *exchange* shortcut rules. Greg's rule is the exception.)

Our first shortcut rule is *Greg's Rule*. We know that if a conjunction contains an atomic formula and the negation of that atomic formula then we can derive the negation of the conjunction. E.g., we can derive the negation of $(K\text{a} \wedge G\text{b} \wedge K\text{c} \wedge \sim G\text{b})$. So if we have a disjunction, one disjunct of which contains a contradiction of this kind, we can derive the negation of that disjunct and use disjunctive syllogism to prune that disjunct. For example, assume the Method ends at the sentence $(K\text{a} \wedge G\text{b} \wedge K\text{c} \wedge \sim G\text{b}) \vee$

$(Ka \wedge Gb \wedge Kc \wedge \sim Ga)$. We can independently derive $\sim(Ka \wedge Gb \wedge Kc \wedge \sim Gb)$. From these two we derive $(Ka \wedge Gb \wedge Kc \wedge \sim Ga)$. Greg's Rule lets us accomplish those steps by crossing out the contradictory part and writing down the remaining ones.

It is helpful to have a short cut rule which combines $\wedge$-*Elim* steps with $\vee$-*Elim* steps to go from a disjunction of which each disjunct contains a particular sentence to that sentence itself on a later line; we call it $\vee/\wedge$-*Elim* and it sanctions the step from $(Ka \wedge Gb \wedge Kc \wedge \sim Gc) \vee (Ka \wedge Gb \wedge Kc \wedge \sim Ga)$ to Gb. In addition to citing the justification, if the sentence is at all complex you should circle the repeated subsentence.

Finally, given the opposite of even one conjunct in a conjunction, we can derive the negation of the conjunction. E.g., from Ga we can derive $\sim(Ka \wedge Gb \wedge Kc \wedge \sim Ga)$. We call this rule *One Bad Apple*, or *OBA*.

Name	**Given**	**May Add**
Greg's Rule	$\psi_1 \vee ... \vee \psi_n$, where some $\psi_i = \phi_1 \wedge ... \wedge \phi_j \wedge ... \wedge \sim\phi_j \wedge ... \wedge \phi_m$	$\psi_1 \vee ... \vee \psi_{i-1} \vee \psi_{i+1} \vee ... \vee \psi_n$
$\vee/\wedge$-*Elim*	$\psi_1 \vee ... \vee \psi_n$, where each ψ_i contains ϕ	ϕ
OBA	$\phi_1 \wedge ... \wedge \phi_i \wedge ... \wedge \phi_n$, $\sim\phi_i$	$\sim(\phi_1 \wedge ... \wedge \phi_i \wedge ... \wedge \phi_n)$
	$\phi_1 \wedge ... \wedge \sim\phi_i \wedge ... \wedge \phi_n$, ϕ_i	$\sim(\phi_1 \wedge ... \wedge \sim\phi_i \wedge ... \wedge \phi_n)$

Table 8.6: Short-Cut Rules for the Method

8.5 Strong Completeness and Other Results

In this section we want to extend our results and show that QD is strongly complete. Note that the method we used to extend the Weak Completeness Theorem to the Completeness Theorem does not work here. To do that, we used theorem 8.12 (on page 186), the proof of which depending on Δ being finite. To show that QD is strongly complete, we have modify the method we used to show that it's weakly complete.

Theorem 8.29. Strong QD Completeness Theorem: For any set Δ of SL sentences and any SL sentence ϕ, if $\Delta \models \phi$, then $\Delta \vdash \phi$.

To show that QD is weakly complete, we gave a method that, given a sentence θ, either produces a derivation of θ or produces a model $\mathfrak{m}$ which makes θ false. To show that QD is strongly complete, what we want is a method that, given a (possibly infinite) set Δ of sentences and another sentence ϕ, either produces a derivation of a contradiction $A \wedge \sim A$ from $\sim\phi$ and some finite subset of Δ or produces a model $\mathfrak{m}$ that makes $\sim\phi$ and every sentence in Δ true.

The method we'll give is a modification of the original method given in section 8.4.3. Since it is just a modification of the the original method, we'll only sketch the changes needed. We'll call the modified method the *strong method*. Given some possibly countably infinite set Δ and sentence ϕ, the strong method is:

Step 1: Let $\Delta^* = \Delta \cup \{\sim\phi\}$. Then pair each sentence of Δ^* with a natural number and use that to determine the order in which they are assumed. The only constraint on this ordering is that $\sim\phi$ should be first.

Step 2: Put the first sentence of Δ^* on line 1 and put it in PDNF, just as was done in Step 0 of the method.

Step 3: Apply Step 1 of the method.

Step 4: Apply Steps 2 and 3 of the method to the whole derivation thus far.

Step 5: Check for contradictions, just as in Step 4 of the method.

Step 6:

(1) If there's a contradiction, stop.

(2) If there's no contradiction, write the next sentence of Δ^* on the next line of the derivation, put that sentence in PDNF, and go back into Step 4.

The strong method either halts in a contradiction, or it does not.

Theorem 8.30. Strong Derivational Lemma: If the strong method halts in a contradiction, then $\Delta \vdash \phi$.

Proof: If the strong method halts in a contradiction, then it has produced a derivation of a contradiction $A \wedge \sim A$ from $\sim\phi$ and some subset Δ' of Δ. We want to show that $\Delta \vdash \phi$; to do this, we first want to show that $\Delta' \vdash \phi$.

We might think we already know that $\sim\phi, \Delta' \vdash A \wedge \sim A$, but in fact, there are additional open assumptions that we made; the Strong Method tells us to make an additional assumption for each line containing an existentially quantified sentence. For these lines, we assume an instance of the existentially quantified sentence. We want to discharge these assumptions by using $\exists$-*Elim*, but these assumptions may come prior to some of the assumptions we made from Δ'. We want to keep the sentences of Δ' as assumptions, because the contradiction we reached depends on them.

We can discharge the assumed instances of existentially quantified sentences only by additionally discharging all the assumptions that come later in the derivation. Accordingly, we want to extend our derivation so that we discharge *all* our assumptions and then repeat all the assumptions *except* for the instances of existentially quantified sentences.

Let $\theta_1, \theta_2, \theta_3, \ldots, \theta_n$ be the sentences of Δ' assumed in our derivation. The last open assumption is either (a) the last sentence of Δ', i.e., θ_n; (b) an instance of an

existentially quantified sentence on an earlier line of the derivation which we'll call ψ; or (c) $\sim\phi$. If (a) is the case, then discharge the assumption by applying the rule $\rightarrow$-*Intro* to get $\theta_n \rightarrow (A \wedge \sim A)$. If (b) is the case, then first apply $\rightarrow$-*Intro* to get $\psi \rightarrow (A \wedge \sim A)$, and then apply $\exists$-*Elim* to get $A \wedge \sim A$. When (b) is the case, we know that the assumption introduced a *new* constant, and therefore we know that that constant did not appear in any earlier assumption or in the existential of which we are taking an instance. It also (obviously) does not occur in $A \wedge \sim A$. Thus the $\exists$-*Elim* step is legitimate. If (c) is the case then we don't have to worry about instances of existentially quantified sentences, and we can skip this part of the process. Let us disregard case (c) for now.

Now we must continue to discharge the rest of the assumptions. For any assumption of an instance of an existentially quantified sentence, we may do the same thing we did in (b) above—first use $\rightarrow$-*Intro* to derive a conditional, and then use $\exists$-*Elim* to derive the RHS of that conditional. For any assumption that is a sentence of Δ' (i.e., θ_i), we use $\rightarrow$-*Intro* to derive a conditional, as in (a) above. So, after discharging the assumption with the second to last sentence from Δ' we get $\theta_{n-1} \rightarrow (\theta_n \rightarrow (A \wedge \sim A))$. After the third instance, we derive $\theta_{n-2} \rightarrow (\theta_{n-1} \rightarrow (\theta_n \rightarrow (A \wedge \sim A)))$. And so on, so that we eventually get a sentence of the form $\theta_1 \rightarrow (\theta_2 \rightarrow (\theta_3 \rightarrow \ldots (\theta_n \rightarrow (A \wedge \sim A))))$.

After discharging all such assumptions, the only open assumption left is $\sim\phi$. Apply $\rightarrow$-*Intro* once more to get something like the following:
$\sim\phi \rightarrow (\theta_1 \rightarrow (\theta_2 \rightarrow \ldots (\theta_n \rightarrow (A \wedge \sim A))))$. Now we have no open assumptions remaining. (Note that we have effectively covered case (c) above, since applying $\rightarrow$-*Intro* in this case would give us $\sim\phi \rightarrow (A \wedge \sim A)$ and no open assumptions. In this case, we didn't have to assume any of the sentences of Δ to get a contradiction, so Δ' is the empty set.)

At this point we have a conditional, possibly a very long one. The RHS of the last conditional (possibly embedded in several conditionals) is our contradiction, $(A \wedge \sim A)$. We want to show that $\Delta' \vdash \phi$, so let us make a series of assumptions from the sentences of Δ'. That is, let us assume each of $\theta_1, \theta_2, \theta_3, \ldots, \theta_n$. Now that we've assumed all the sentences of Δ', let us assume $\sim\phi$.

Given our earlier conditional of the form $\sim\phi \rightarrow (\theta_1 \rightarrow (\theta_2 \rightarrow \ldots, (\theta_n \rightarrow (A \wedge \sim A))))$ and the assumption $\sim\phi$, we may now apply $\rightarrow$-*Elim* to get
$\theta_1 \rightarrow (\theta_2 \rightarrow (\theta_3 \rightarrow \ldots, (\theta_n \rightarrow (A \wedge \sim A))))$. Because we also have all the sentences of Δ' as assumptions (i.e., all of $\theta_1, \theta_2, \theta_3, \ldots, \theta_n$), we may apply a series of $\rightarrow$-*Elim* steps until we eventually derive $A \wedge \sim A$. That is, given our earlier assumption θ_1 and the conditional $\theta_1 \rightarrow (\theta_2 \rightarrow (\theta_3 \rightarrow \ldots, (\theta_n \rightarrow (A \wedge \sim A))))$, we may apply $\rightarrow$-*Elim* to derive $\theta_2 \rightarrow (\theta_3 \rightarrow \ldots, (\theta_n \rightarrow (A \wedge \sim A)))$. And then because we have θ_2 as an assumption we may again apply $\rightarrow$-*Elim* to get $\theta_3 \rightarrow \ldots, (\theta_n \rightarrow (A \wedge \sim A))$. And so on, until we derive $A \wedge \sim A$.

Remember that our last open assumption is $\sim\phi$. We may now discharge that assumption and apply $\rightarrow$-*Intro* to get $\sim\phi \rightarrow (A \wedge \sim A)$. Then we apply $\sim$-*Elim* to derive ϕ.

Now we have as our open assumptions only the sentences of Δ' and we have derived

ϕ. We have thus shown that $\Delta' \vdash \phi$. And because Δ' is a subset of Δ, it follows that $\Delta \vdash \phi$. ■

Next, note that if the strong method doesn't halt in a contradiction, then we have a list of matrix instances from which we can construct a matrix model $\mathfrak{m}_M$ in just the same way we did for the method (Def. 8.22, on page 198). Similar to the method, we have the following three theorems.

Theorem 8.31. The Strong Method Lemma 1: The matrix model $\mathfrak{m}_M$ makes true all sentences on the master matrix list M.

Proof: The same proof used for the method (Thm. 8.23, on page 199) applies here too. ■

Theorem 8.32. The Strong Method Lemma 2: All matrix instances in the derivation are true in the matrix model $\mathfrak{m}_M$.

Proof: The same proof used for the method (Thm. 8.24, on page 199) applies here too. ■

Theorem 8.33. The Strong Method Lemma 3: All quantified sentences in the derivation are true in the matrix model $\mathfrak{m}_M$.

Proof: The same proof used for the method (Thm. 8.25, on page 199) applies here too, so long as we can show that if an existential $\exists\alpha\psi$ appears on a line, at least one instance $\psi t/\alpha$ does, and if a universal $\forall\alpha\psi$ appears on a line, then every instance $\psi t/\alpha$ of it with a constant t appearing somewhere in the derivation appears somewhere in the derivation. So, all we need to show is that the strong method derives the appropriate instances of all quantified sentences that appear in our derivation. Let Δ' be those sentences of Δ^* that appear in our derivation as a result of the application of the strong method.

Existential Quantifier: Say θ is some sentence in the derivation of the form $\exists\alpha\phi$. Step 4 of the strong method uses step 3 of the method on θ, which guarantees that the sentence $\phi t/\alpha$, for some constant t, appears somewhere in the derivation. By hypothesis, step 4 of the strong method is applied to all sentences in Δ', so we know that it derives the appropriate instances of all existentially quantified statements in in Δ'.

Universal Quantifier: Say θ is a sentence appearing in the derivation of the form $\forall\alpha\phi$. Step 4 of the strong method uses step 2 of the method on θ, which, for every constant t in the derivation, guarantees that the sentence $\phi t/\alpha$ is derived. By hypothesis, step 4 of the strong method is applied to all sentences in Δ', so we know that it derives the appropriate instances of all universally quantified statements in Δ'.

So, the strong method derives the appropriate instances of all quantified sentences in Δ'.

■

Finally, we have one last lemma:

Theorem 8.34. Main Strong QD Completeness Lemma: For all sets of QL sentences Δ and QL sentences ϕ, if the strong method is applied to $\Delta^* = \Delta \cup \{\sim\phi\}$ then either: (a) the strong method produces a derivation of ϕ from Δ in QD^+, or (b) there is a model $\mathfrak{m}$ which makes every sentence in Δ true and ϕ false.

Proof: If the method is applied to $\Delta^* = \Delta \cup \{\sim\phi\}$, then either it halts in a contradiction or not. By the Strong Derivational Lemma (8.30, on page 203), if the strong method halts in a contradiction, then $\Delta \vdash \phi$ in QD^+.

If the method does not halt in a contradiction, then by the Strong Method Lemma 3 (Thm. 8.33, on the previous page) the matrix model $\mathfrak{m}_M$ makes all the sentences in the derivation true. But since the strong method did not halt in a contradiction, for every sentence ψ in Δ and $\sim\phi$, there's some sentence in PDNF that's quantificationally equivalent to ψ and appears in the derivation. So $\mathfrak{m}_M$ makes all the sentences in Δ and the sentence $\sim\phi$ true; hence $\mathfrak{m}_M$ makes all the sentences in Δ true and ϕ false. ■

Proof of Thm. 8.29, The Strong Completeness Theorem for GQD: Assume that $\Delta \models \phi$. Then there can be no model $\mathfrak{m}$ which makes all of the sentences in Δ true and ϕ false; so, application of the method can't produce such a model. Thus by the Main QD Strong Completeness Lemma (Thm. 8.34), $\Delta \vdash \phi$ in QD^+. It follows by theorem 7.3 (on page 171) that $\Delta \vdash \phi$ in QD. ■

Our Method for proving completeness for QD is short of being a decision procedure. If ϕ is a logical truth, then The Method produces a derivation of it. And if ϕ is not a logical truth, then in many cases it produces a model that makes ϕ false. But sometimes it just doesn't stop. We know that if it doesn't stop there is a model that makes the original sentence false, but at each stage we may not know whether it stops (soon?) or not. And we know that it can't stop at a finite stage for some sentences because those sentences are only false in an infinite model.

All we know from our work so far is that The Method works as described above. We don't know that there isn't a better method that provides a decision procedure. Church's Theorem, proved by more advanced methods that involve clarifying what counts as a "method" or "algorithm" tells us that our result is as good as we can do for all of QL.

However, we can do better for the language QL1 and a little more.

8.6 Decidability and Church's Theorem

Next we turn to refinements of the method to obtain what are called *decision procedures* for logical truth in a language L. We first introduced the idea of a decision

procedure in section 7.1.1 (on page 169); here we shall fill things out a bit further.

Definition 8.35. A *decision procedure* for logical truth in a language (or sublanguage) L is a completely specified method which produces, for any sentence ϕ of L and in a finite number of steps, the answer YES if ϕ is a logical truth and the answer NO otherwise.

Example 8.36. We have already seen two decision procedures for TFT in SL: truth tables and the method discussed in the completeness proof of SD. (Others include truth trees and Quine's "fell swoop", Quine 1950, Hodges 2001b: 23.) The truth-table decision procedure is simple: take a sentence ϕ of SL and construct a truth table for it. If you get all T in the column under ϕ; answer YES. If you don't, answer NO. Likewise for the method discussed in the completeness proof of SD: take a sentence ϕ, negate it to get $\sim\phi$, and apply the method. If it results in a contradiction $A \wedge \sim A$, answer YES. If no contradiction is reached, answer NO.

Our method of proving completeness for QD is a little short of being a decision procedure for quantificational truth in QL because it does not always produce an answer in a finite amount of time. It can be shown that there is no decision procedure for the whole language QL (Hodges 2001b: 83–86, 2001a: 31, Bergmann et al. 2003: 486).

Theorem 8.37. Church's Theorem: If L is a sublanguage of QL with (1) the same logical connectives as QL, and (2) at least one 2-place predicate symbol, then there is no decision procedure for the set of logical truths of L.[7]

Church's Theorem at once tells us that there is no decision procedure for quantificational truth in QL, but we can show that with some modifications our method from section 8.4.3 can be turned into a decision procedure for certain sublanguages of QL. As the theorem suggests, one such sublanguage of QL consists of just 1-place predicate symbols. We've been calling this language QL1.

Let's modify the method to prove that QL1 has a decision procedure. The problem with the existing method is that infinite loops are created by having an existential quantifier inside a universal quantifier. The existential requires a new constant to be instantiated, and that creates a new potential instance for the universal, which then creates a new existential, and so on. We want a procedure that is guaranteed to halt.

If the method produces a sentence in standardized form which has only existential, or only universal quantifiers, then the method stops. Moreover, if in the standardized form the existentials all precede the universals the method also stops, for we first take instances of all the existentials using n constants if there are n existentials, and afterwards we instantiate the n new constants in the m universals giving n^m instances. We are done then, since no additional constants are added.

[7] Actually, Church's Theorem also says that if we also consider languages with function symbols, then if L has at least two 1-place function symbols there is no decision procedure for the set of logical truths of L.

But what about sentences with an existential quantifier within the scope of a universal quantifier, e.g. $\forall x \exists y((Bx \wedge Cy) \vee (Dx \wedge Gy))$? Since QL has many-place predicates, if one quantifier occurs within the scope of another then it often cannot be moved in front of the other quantifier. But since the predicates of QL1 are 1-place there is a procedure to switch quantifier order in all cases. It takes some work to establish this result.

Definition 8.38. Two quantifiers are *independent* iff neither is in the scope of the other.

Let's say we have a sentence all of whose quantifiers are independent of each other. Then these quantifiers can be brought forward in any order, and so we have a decision procedure for such sentences. We claim that each QL1 sentence is equivalent to one whose quantifiers are all independent. To simplify the proof for this we introduce some additional exchange rules.

8.6.1 Additonal Exchange Rules

Our aim is to move the quantifiers of a QL1 sentence so that all are independent. To simplify this task we introduce new quantifier exchange rules, as well as some rules that permit rearranging the order of conjunctions and disjunctions. These rules allow us to provide a procedure to push each quantifier in far enough so that the scope of each covers the variables it binds and no others.

Name	**Given**	**May Add**
$\forall$-*Shuffle*	$\forall\alpha(\phi_1 \wedge ... \wedge \phi_n)$	$\forall\alpha\phi_1 \wedge ... \wedge \forall\alpha\phi_n$
$\exists$-*Shuffle*	$\exists\alpha(\phi_1 \vee ... \vee \phi_n)$	$\exists\alpha\phi_1 \vee ... \vee \exists\alpha\phi_n$
Q-Deletion	$\#\alpha\phi$	ϕ, iff α does not occur in ϕ
$\wedge$-*Shuffle*	$\phi_1 \wedge ... \wedge \phi_k \wedge ... \wedge \phi_n$	$\phi_k \wedge \phi_1 \wedge ... \wedge \phi_n$
$\vee$-*Shuffle*	$\phi_1 \vee ... \vee \phi_k \vee ... \vee \phi_n$	$\phi_k \vee \phi_1 \vee ... \vee \phi_n$

Table 8.7: Additonal Exchange Rules for QD

It is left as an exercise for the reader to prove that these rules are sound. Note that $\forall$-Shuffle is only sound when the universal quantifier is moved over a conjunction, and $\exists$-Shuffle is only sound when moved over a disjunction. For example, the sentence $\forall x(Bx \vee \sim Bx)$ is a logical truth but $\forall xBx \vee \forall x \sim Bx$ isn't. Similarly, $\sim\exists x(Bx \wedge \sim Bx)$ is a logical truth but $\sim(\exists xBx \wedge \exists x \sim Bx)$ isn't.

These rules enable us to push the quantifiers all the way into a suffiently well-behaved sentence. To push universal quantifiers inward over conjunctions, we use $\forall$-*Shuffle*. To push exstential quantifiers inward over disjunctions, we use $\exists$-*Shuffle*. In

cases when the universal quantifer governs a disjunction we must isolate the disjuncts with a matching variable, using $\vee$-*Shuffle* to move those disjuncts to the left. Then we use Q shuffle to move the quantifier to just those disjuncts. Similarly, when an existential quantifier governs a conjunction, we isolate the conjuncts with a matching variable, using $\wedge$-*Shuffle* to move those conjuncts to the left. Then we use Q shuffle to move the quantifier to just those conjuncts.

8.6.2 The QL1 Decision Procedure

Theorem 8.39. QL1 Independent Quantifiers Theorem: Every sentence of QL1 is quantificationally equivalent to a sentence whose quantifiers are independent.

Proof:

The basic strategy to prove this theorem is relatively straightforward even if some of the details aren't.

First, we start with a QL1 sentence and get its PDNF equivalent. All the quantifiers are in the initial position of the latter sentence. Then we push each quantifier inward, using the new quantifier exchange rules, to shrink its scope. We show that after the quantifiers are pushed in they are all independent.

Let there be a sentence of QL1. Then by theorem 8.19 there is an equivalent sentence, ϕ, that is in prenex disjunctive normal form. So ϕ is of the form $\#_1\#_2 ... \#_n\phi'$, where for each $i \in \{1, ..., n\}$, $\#_i$ is a quantifier, and ϕ' is a DNF formula. Then ϕ' is of the form $(\psi_1 \vee \psi_2 \vee ... \vee \psi_k)$, where for each $j \in \{1, ..., k\}$, ψ_j is a conjunction of ions.

Let ϕ_{n+1} be ϕ', i.e., the DNF body of ϕ. To make the quantifiers independent of each other, each quantifier must be pushed in to govern the variables it binds and no others. We start with $\#_n$, pushing the quantifier in, and work backward in sequence to $\#_1$. The formula we start with, ϕ_{n+1}, is transformed n times, once for each of the n quantifiers. Let $\#_i$ be the the i^{th} quantifier of the sequence in ϕ, $\#_1\#_2 ... \#_n$. Then let ϕ_i for each $i \in \{1, ..., n\}$ be the result of having pushed in the i^{th} quantifier, $\#_i$, into the formula ϕ_{i+1}. The final product of all these transformations is ϕ_1, at which point all the quantifiers are moved in. Each transformation uses exchange rules only, so ϕ_1 is equivalent to ϕ.

The following steps define the transformation to push in each quantifier, starting with $\#_n$ and working back to $\#_1$. For each quantifer $\#_i$ to move there are two cases, handled separately:

Case 1: $\#_i$ is an existential quantifier.

To push an existential quantifier in is easier, since the body of the formula is a disjunction. We push it in using ∃-*Shuffle* so that each disjunct is governed by an existential quantifer. Then, for each disjunct, we reorder the conjuncts using ∧-*Shuffle* to isolate the conjuncts with a matching variable, and use Q shuffle to move the quantifier to govern just those.

Step A: By ∃-shuffle, $\#_i$ is moved to each disjunct in ϕ_{i+1}, resulting in: $(\#_i\psi_1 \vee \#_i\psi_2 \vee ... \vee \#_i\psi_k)$. For any ψ_j not containing the variable in $\#_i$, $\#_i$ may be removed from that disjunct, by Q-deletion.

Step B: Each disjunct with a quantifier $\#_i\psi_j$ is of the form $\#_i(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_m)$ where each θ is an ion (or a quantified subsentence governing a conjunction or disjunction, as will be made clear shortly). An equivalent conjunction is obtained by applying ∧-*shuffle* so that each θ with a variable matching that of $\#_i$ is moved to the left.

Step C: By the Q shuffle exchange rule, the $\#_i$ is moved in to govern just the leftmost conjuncts with a matching variable, resulting in each $\#_i\psi_j$ becoming: $\#_i(\theta_1 \wedge \theta_2) \wedge ... \wedge \theta_m$. With this transformation $\#_i$ binds all variables within its scope, and no other variables are in its scope.

The result is a formula ϕ_i where $\#_i$ binds all and only its variables.

Case 2: $\#_i$ is a universal quantifier.

To push a universal quantifier in, the distribution exchange rule is first used to make the body of the sentence a conjunction. Then we push the universal in using ∀-*Shuffle* so that each conjunct is governed by a universal quantifer. Then, for each conjunct, we reorder the disjuncts using ∨-*Shuffle* to isolate the disjuncts with a matching variable, and use Q shuffle to move the quantifier to govern just those. After the quantifier is moved in, the distribution exchange rule is applied again to make the body of the sentence a disjunction again.

Step A: Apply the distribution rule to ϕ_{i+1} until no conjunction is governed by a disjunction (with the exception of any conjunction within the scope of an already moved quantifier). The resulting formula, ϕ^*_{i+1}, is a conjunction.

Step B: By ∀-shuffle, $\#_i$ is moved to each conjunct in ϕ^*_{i+1}, resulting in: $(\#_i\psi_1 \wedge \#_i\psi_2 \wedge ... \wedge \#_i\psi_k)$. For any ψ_j not containing the variable bound by $\#_i$, the quantifier may be removed, by Q-deletion.

Step C: Each conjunct with a quantifier $\#_i\psi_j$ is of the form $\#_i(\theta_1 \vee \theta_2 \vee ... \vee \theta_m)$ where each θ is an ion (or a quantifier subsentence governing a disjunction or conjunction). An equivalent disjunction is obtained by by applying $\vee$-*shuffle* so that each θ with a variable matching that of $\#_i$ is moved to the left.

Step D: By the Q shuffle exchange rule, the $\#_i$ is moved in to govern just the leftmost disjuncts with a matching variable, resulting in $\#_i\psi_j$ becoming: $\#_i(\theta_1 \vee \theta_2) \vee ... \vee \theta_m$. With this transformation $\#_i$ binds all variables within its scope, and no other variables are in its scope.

Step E: Apply the distribution rule to the resulting formula until no disjunction is governed by a conjunction (with the exception of any disjunction within the scope of an already moved quantifier).

The result is a formula ϕ_i where $\#_i$ binds all and only its variables.

The result of these n transformation is a QL1 sentence ϕ_1 with each moved quantifier governing all and only the variables it binds. Therefore the quantifiers of ϕ_1 are all independent. ■

Theorem 8.40. The QL1 Decision Theorem:

Proof: Let ϕ be a QL1 sentence. Then by theorem 8.39 there is an equivalent sentence ϕ^* such that all its quantifiers are independent. Use Q-shuffle to move the existential quantifers to the prenex position, and then move the universal quantifiers as well, resulting in an equivalent sentence ϕ^{**}. Then put the body of ϕ^{**} into DNF, resulting in ϕ^{***}. ϕ^{***} is in PDNF. So for the reasons described previously, the method is guaranteed to halt on ϕ^{***}. Therefore there is a decision procedure for ϕ^{***}, and hence also for ϕ. ■

8.7 Löwenheim-Skolem and Compactness

A number of results follow directly from the completeness of QD or the method we used to prove completeness.

Theorem 8.41. The Downward Löwenheim-Skolem Theorem: If a sentence of QL is true in any model, then it is true in one whose domain consists of all or some of the natural numbers.

Proof: If ϕ is true in some model, then $\sim\phi$ is not a quantificational truth. Thus, applying the method to $\sim\phi$ does not produce a contradiction, but produces a model of the natural numbers which falsifies $\sim\phi$ and hence makes ϕ true. ■

It's important to note that there's really nothing special about the natural numbers. When we devised the procedure for constructing a model of the ions in the master

matrix list that results from the method (when no contradiction arises), we choose to use natural numbers for the universe. But it should be clear that we did this out of convenience (it's easy, after all, to associate constants with the natural numbers). We could have used any set of objects for the universe. What's important is that, whatever we used, the domain of the constructed model is at most countably infinite (i.e., it is at most the size of the natural numbers and no larger). Hence a more abstract version of the downward Lowenheim-Skolem Theorem simply says: If a sentence of QL is true in any model, then either it's true in only models with finite domains, or, if it's true at all in models with infinite domains, then there's an model with a *countably* infinite domain in which it's true.

This thorem was proved in a weaker form originally by Leopold Löwenheim (1915), and the proof was improved by Thoralf Skolem (1920; 1922). Notice that the theorem talks only about models, and we have proved it via a detour through derivations. As you might imagine, there are more direct proofs, including Skolem's (Tarski and Vaught 1956, Vaught 1974, Hodges 1997: ch. 3.1, 2001b: 63).

Notice also that after our work on QL1, we know that for monadic sentences the method stops after a finite number of steps (if we arrange the prenex carefully) and so we can conclude that if a monadic sentence is true in any model then it is true in a finite one.

Theorem 8.42. If ϕ is a sentence of QL1 and has a model, then it has a finite model.

Our next corollary of completeness is the Compactness Theorem. Although historically the completeness theorem was proved first and compactness followed as a corollary, today the compactness theorem takes center stage in many areas of logic (especially model theory). Like the Löwenheim-Skolem Theorem, there are many different proofs of compactness that do not go through completeness or use any facts about derivations (see Kleene 2002 [1967]: 321, Ebbinghaus 1985, Hodges 1997: ch. 5.1, Hodges 2001b: 63, 2001a: 29).

Theorem 8.43. The Compactness Theorem for QL: For all sets of sentences Δ of QL, if for every finite subset Δ' of Δ there exists a model $\mathfrak{m}'$ that makes all the sentences in Δ' true, then there's some model $\mathfrak{m}$ that makes all the sentences in Δ true.

Proof: By the strong completeness theorem, for all sets Δ of QL sentences and QL sentence ϕ, if $\Delta \models \phi$, then $\Delta \vdash \phi$. Now assume that there's no model $\mathfrak{m}$ that makes all the sentences in Δ true. Hence $\Delta \models A \wedge \sim A$. So by strong completeness, $\Delta \vdash A \wedge \sim A$. By definition, this implies that there's some finite subset Δ' of Δ such that $A \wedge \sim A$ can be derived from Δ'. Hence there is no model $\mathfrak{m}$ that makes all the sentences in Δ' true. Hence it's not the case that for every finite subset Δ' of Δ there exists a model $\mathfrak{m}'$ that makes all the sentences in Δ' true. ■

8.8 Exercises

8.8.1 Misc. Problems

1. Let's say that any derivation rule R that has the following property is *sound*: if we add a line to a derivation D with sentence ϕ sanctioned by rule R, then $\Delta \models \phi$, where Δ is the set of unboxed assumptions for the new line. (Compare this with what it is for a rule to be truth-preserving, def. 8.1 on page 179, which is different.) Then the proof of theorem 8.5 (on page 180) basically shows that SD is sound by showing that all the basic rules of SD are sound. We know that SD^+ is sound because SD is sound and (by theorem 6.7, on page 148) anything you can derive in SD^+ can be derived in SD. But we could also show that SD^+ is sound directly (without appealing to theorem 6.7) by showing that the shortcut rules used in SD^+ themselves are sound. Of course, this follows from theorem 6.6, on page 148 and the fact that the basic rules are sound, but again we can show it directly. But again we can show it without going through the basic rules. Show directly (without appealing to theorem 6.6) that the following rules are sound (see tables 6.39, on page 145 and 6.40, on page 146):

   (a) *M.T.*
   (b) *A.C.*
   (c) $\rightarrow/\vee$*-Exch.*
   (d) *Contraposition*

2. Recall that $\rightarrow$ elimination can only be used on a conditional that is the main connective of a sentence. Show that if we do not make this restriction, then the rule is unsound. In other words, give a derivation which violates only that restriction (a derivation where you use $\rightarrow$ elimination on a horseshoe that's not the main connective) and which ends with a proof of a sentence that is *not* a logical truth (not truth-functionally true) from the empty set of assumptions.

8.8.2 Quantifier Exchange Rule Soundness

Prove the soundness of the following exchange rules.

1. $\forall$*-Shuffle*
2. $\exists$*-Shuffle*
3. *Q-Deletion*
4. $\wedge$*-Shuffle*
5. $\vee$*-Shuffle*

Name	**Given**	**May Add**
$\forall$-*Shuffle*	$\forall\alpha(\phi_1 \wedge ... \wedge \phi_n)$	$\forall\alpha\phi_1 \wedge ... \wedge \forall\alpha\phi_n$
$\exists$-*Shuffle*	$\exists\alpha(\phi_1 \vee ... \vee \phi_n)$	$\exists\alpha\phi_1 \vee ... \vee \exists\alpha\phi_n$
Q-Deletion	$\#\alpha\phi$	ϕ, iff α does not occur in ϕ
$\wedge$-*Shuffle*	$\phi_1 \wedge ... \wedge \phi_k \wedge ... \wedge \phi_n$	$\phi_k \wedge \phi_1 \wedge ... \wedge \phi_n$
$\vee$-*Shuffle*	$\phi_1 \vee ... \vee \phi_k \vee ... \vee \phi_n$	$\phi_k \vee \phi_1 \vee ... \vee \phi_n$

Table 8.8: Additonal Exchange Rules for QD

Chapter 9

Further Directions

9.1 Many-Valued Logic

In previous chapters we assumed that there are only two truth values. We now set aside that assumption to explore many-valued logics. We consider the possibility that there are varying degrees of truth and falsity, such that there are sentences that are neither wholly true nor wholly false. Among the reasons that have been given historically for rejecting the two-valuedness assumption are the beliefs that statements about the future, statements involving vague predicates, or statements about quantum-mechanical properties are not always either true or false. Most many-valued logics begin by rejecting the law of excluded middle $A \vee \sim A$, though there are exceptions.[1] The number of values ranges from three to infinity. The interpretations of the further values vary widely from one author to another, as do the motivations for introducing the additional values.

Emil Post was one of the first to study many-valued logics, but his motivation seems to have been entirely formal. The other major founder of many-valued logic was Łukasiewicz. He sketched the idea of a many-valued logic in 1920 and published a systematic account in 1930.[2] Unlike Post, Łukasiewicz introduced three-valued logic for a philosophical reason: to provide a more appropriate representation for the indeterminacy of the future. He apparently was led to this both by a historical concern—studying Aristotle's discussion of necessity, particularly his sea battle example—and by a quite contemporary concern about how to accommodate the indeterminism of modern physics within logic. Aristotle's sea battle argument is as follows:

(1) If there will be a sea battle tomorrow, then necessarily there will be a sea battle tomorrow.

(2) If there will not be a sea battle tomorrow, then necessarily there will not be a sea battle tomorrow.

(3) Either there will or there will not be a sea battle tomorrow.

[1] For example, the most generally accepted logic for quantum mechanics keeps $A \vee \sim A$ but rejects one of the distribution principles.

[2] Both are reprinted in Borkowski 1970.

(4) Therefore, either there will necessarily be a sea battle tomorrow or there will necessarily not be a sea battle tomorrow.

Aristotle suggested that the third premise—the law of excluded middle, $A \vee {\sim}A$—should be rejected when A is a statement about a future contingency. Thus the motivation, if not the details, of many-valued logic is as ancient as the study of logic itself. Łukasiewicz developed this idea into a systematic logic.

In all of his later work, Łukasiewicz used 1 for truth, 0 for falsity, and intermediate values for other truth values. Most but not all writers use this convention. However, it is one thing to decide that $1/2$ is your third truth value and another thing to give a philosophical explanation of it. For Łukasiewicz the intermediate value is "indeterminate". Given this understanding, the most natural three-valued generalization of the two-valued truth tables is the following, in which negation reverses the truth value.

A	${\sim}A$
1	0
½	½
0	1

Conjunction takes the minimum value of the conjuncts, while disjunction takes the maximum value of the disjuncts.

$A \wedge B$	1	½	0
1	1	½	0
½	½	½	0
0	0	0	0

$A \vee B$	1	½	0
1	1	1	1
½	1	½	½
0	1	½	0

For example, the conjunction of a true sentence and an indeterminate sentence is indeterminate. It is instead true if the indeterminacy is resolved in favor of truth and false if the indeterminacy is resolved in favor of falsity.

Note that when all the components of a sentence formed from these connectives are assigned value $1/2$ the entire sentence has value $1/2$. If the conditional $A \to B$ is defined as equivalent to as ${\sim}A \vee B$, as is often done in two-valued logic, then it too is

indeterminate when A and B are. As a consequence $A \to A$ is equivalent to the law of excluded middle, $\sim A \vee A$, and $A \to A$ is not a logical truth. This outcome is generally thought undesirable.

Instead of using that traditional, oft questioned equivalence, Łukasiewicz defined the conditional thus:

$A \to B$	1	½	0
1	1	½	0
½	1	1	½
0	1	1	1

One way of describing this table is that the conditional is false only in the case of $T \to F$ and is indeterminate only in two cases: $T \to I$ and $I \to F$. A rationale for these choices is that if A were true and B indeterminate, then the conditional $A \to B$ could be true if B were to be true and false if B were to be false. The choice of the value 1 when both constituents have value $1/2$ is required if $A \to A$ is to be logically true.

Equivalence can be defined as usual: $A \leftrightarrow B$ iff $A \to B$ and $B \to A$. The truth table for the biconditional is specified by this definition above. We leave it to the reader to give a simpler, more direct explanation of the truth table. In Łukasiewicz' presentation of his system, he used only negation and the conditional, having noted that $A \vee B$ can be defined as $(A \to B) \to B$ and then $A \wedge B$ can be defined by using the usual DeMorgan's principle.

In two-valued logic a sentence is logically true iff it is true in all models. When there are more than two truth values, we must indicate which subset of the values are the designated values, i.e., those that are truth-like. The definition is revised: A is a *logical truth* iff it has a designated value in all models.

Since Łukasiewicz' motivation was to deny excluded middle he chose only 1 as a designated value. This achieves the purpose of rendering excluded middle not a logical truth. It has one somewhat counterintuitive consequence though, which is that under a model in which both constituents are assigned value $1/2$, $A \wedge \sim A$ has the same truth value as $A \vee \sim A$. Issues of the indeterminacy of the future are now generally studied within the framework of tense logic. Łukasiewicz' innovations have opened the possibilities for a variety of other systems and ideas.

9.1.1 Finite-valued systems with more than three values

The Łukasiewicz three-valued generalization can be systematically carried further. The n-valued generalization consists of taking the values $i/n-1$ for $0 \leq i \leq n-1$. For example, four-valued Łukasiewicz logic has the values 0, $1/3$, $2/3$, and 1.[3] Conjunction

[3]Four-valued logic was proposed for modal logic, the values being "necessarily true", "contingently true", "contingently false", and "necessarily false". The Łukasiewicz definitions of the usual connectives can be used and a modal operator added. While these truth tables have some uses, they have

takes the minimum value of the conjuncts, and disjunction the maximum value of the disjuncts. The value of a negation is 1 minus the value of the sentence being negated. For the conditional $A \rightarrow B$ we have two cases, using $V(A)$ for the value of A:

$$V(A \rightarrow B) = \begin{cases} 1 & \text{if } V(A) \leq V(B) \\ [1\text{–}V(A)] + V(\mathrm{B}) & \text{otherwise} \end{cases}$$

In all of the Łukasiewicz systems the only designated value is 1. Excluded middle is not logically true in any of these systems, though in the even-valued systems excluded middle is always truer than the contradiction $A \wedge \sim A$.[4]

9.1.2 Infinite-valued systems

The Łukasiewicz n-valued generalization can be systematically carried further still. Łukasiewicz also studied the cases where the set of truth values consists of all rational numbers, or all the real numbers, in the interval $[0, 1]$. Conjunction, disjunction, negation, and the conditional (with the two cases) are the same as for finite-valued systems with more than three values. The set of logical truths in the three-valued logic is a subset of those in our traditional logic; if a sentence can be shown to be false in a two-valued model, then that model also falsifies the sentence in the three-valued system.

One question to ask is whether all these values make a difference. We already know part of the answer. $A \vee \sim A$ is a two-valued logical truth but not a three-valued one. But this does not give us an answer for the other systems. Before addressing this question, let's generalize our observation about excluded middle. If we think about the truth tables for $\sim$, $\wedge$, and $\vee$, we can see that when the input value(s) are $1/2$, the output value is always $1/2$. Thus a simple recursive proof (that we won't bother giving) shows that, in a model in which all atoms receive value $1/2$, any sentence without conditionals and biconditionals is not a logical truth, because it receives value $1/2$ on that model.

Theorem 9.1. All logical truths of Łukasiewicz systems contain conditionals or biconditionals.

This means that if we are looking for sentences that discriminate between the n-valued systems then we need to look at conditionals or biconditionals. One candidate for a form of sentence is $(A \rightarrow B) \vee (B \rightarrow C)$. This is a two-valued logical truth but not a three-valued one, because we can assign the values 1, $1/2$, and 0 to the respective atoms and give the sentence a value of $1/2$. Generalizing, sentences of the form

$$(A_1 \rightarrow A_2) \vee (A_2 \rightarrow A_3) \vee (A_3 \rightarrow A_4) \vee \ldots \vee (A_n \rightarrow A_{n+1})$$

been superseded by the possible worlds approach to modal logic. We discuss possible world semantics later in this chapter.

[4]Systems with more than one designated value were mentioned by Post, and this variation on Łukasiewicz systems was studied by Slupecki and others.

are logical truths in an n-valued system but not in a system with at least $n+1$ values.

Example 9.2. Is the sentence $(A \wedge \sim B) \rightarrow \sim(A \rightarrow B)$ a three-valued truth? If so, are they n-valued truths for all finite n?
Solution. This is not a logical truth in any system with more than two values. If we assign both A and B the same intermediate value, then $\sim B$ has an intermediate value and so $A \wedge \sim B$ has an intermediate value. But since A and B have the same value, $A \rightarrow B$ has the value 1 and $\sim(A \rightarrow B)$ has the value 0. So, the whole conditional has an intermediate value, not 1.

Example 9.3. Is the sentence $((A \rightarrow C) \wedge (B \rightarrow C)) \rightarrow ((A \vee B) \rightarrow C)$ a three-valued truth? If so, are they n-valued truths for all finite n?
Solution. We give an argument that this sentence is true for all finite-valued logics. If the values of A and B are both less than or equal to that of C, then the right conditional has value 1 and consequently the whole sentence does too. Thus it can only be less than 1 if A or B has value greater than C. Suppose A is greater than C and greater than or equal to B. Then $\text{Max}(A, B) = A$ and the right conditional has the same value as $A \rightarrow C$. But since $A > B$, $1 - A < 1 - B$, and so the value of the left conditional is also the value of $A \rightarrow C$ because that is the minimum value on that side.

9.2 Modal Sentential Logic

We can think of sentential logic as the logic of truth functional operators, and quantificational logic as the logic of quantifiers and predicates. But there is a lot left out by this: possibility and necessity, deontic concepts like permissibility and obligation, doxastic concepts like belief and knowledge, and temporal concepts like past and future, just to name a few. Logics which attempt to capture these concepts are called modal logics. Sometimes the term 'modal logic' refers just to the logic of possibility and necessity, and at other times it refers to the broader class of modal logics just mentioned. In this section we give a brief introduction to the logic of possibility and necessity.

9.2.1 The Language MSL

The language of modal sentential logic, MSL, is an extension of SL. We add to SL two 1-place logical connectives: $\Box$ and $\Diamond$. Intuitively, $\Box\phi$ means that ϕ is necessarily true, while $\Diamond\phi$ means that ϕ is possibly true.

Definition 9.4. The *sentences of MSL* are defined recursively:

Base Clause: Every sentence letter (def. 2.1) is a sentence.

Generating Clauses:

(1) If ϕ is a sentence, then so are $\sim\phi$, $\Box\phi$ and $\Diamond\phi$.

(2) If ϕ and θ are sentences, then so are $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.

(3) If all of $\phi_1, \phi_2, \phi_3, \phi_4, \ldots, \phi_n$ are sentences (the list must include at least two sentences and be finite), then so are $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \ldots \vee \phi_n)$.

Closure Clause: A sequence of symbols is a MSL sentence if and only if its being a sentence follows from the previous two clauses.

Of course, as with SL, sentences of MSL only count as either true or false relative to a model.

9.2.2 Truth, Logical Truth, and Entailment in MSL

Like the models for QL (and unlike the many-valued models for SL), models for MSL can be thought of as extensions of the (2-valued) models of SL (recall definition 2.18, on page 22).

Definition 9.5. A model $\mathfrak{m}$ for MSL is an ordered triple $\langle w_0, W, V \rangle$ where:

(1) W is a set, the elements of which are thought of as "possible worlds"

(2) $w_0 \in W$, where w_0 plays the role of "the actual world"

(3) V is a function that assigns a subset of W to each sentence letter of MSL.

Elements of W, "possible worlds", are denoted by w, w_0, w_1, w_2 and so on. The subset $V(\phi)$ of W assigned to a sentence letter ϕ can be the empty set $\emptyset$, a non-empty proper subset of W, or all of W itself. Intuitively, the set $V(\phi)$ is the set of possible worlds in which ϕ is true. And $W - V(\phi)$, those worlds not in $V(\phi)$, are worlds in which ϕ is false. We use the '$-$' symbol to indicate the complement of a set.[5] Thus the function V in a MSL model is like a SL model, except that it assigns truth values to sentence letters in a possible world. Just as before, we can extend the notion of truth in a possible world w on a model $\mathfrak{m}$.

Definition 9.6. The following clauses define a function V^* which extends V to all sentences of MSL.

(1) If ϕ is a sentence letter, $V^*(\phi) = V(\phi)$.

(2) If ϕ is $\sim\psi$, then $V^*(\phi) = W - V(\psi)$, i.e. $V^*(\phi)$ is the set of worlds in W not in $V(\psi)$.

(3) If ϕ is $(\psi_1 \wedge \ldots \wedge \psi_n)$, then $V^*(\phi) = V^*(\psi_1) \cap \ldots \cap V^*(\psi_n)$.

(4) If ϕ is $(\psi_1 \vee \ldots \vee \psi_n)$, then $V^*(\phi) = V^*(\psi_1) \cup \ldots \cup V^*(\psi_n)$.

(5) If ϕ is $(\psi \rightarrow \theta)$, then $V^*(\phi) = V^*(\sim\psi) \cup V^*(\theta)$.

[5] For example, relative to some domain, $\{1, 2, 3, 4\}$, the complement of the set $\{1, 2\}$ is $\{3, 4\}$. We indicate the latter with the following notation: $\{1, 2, 3, 4\} - \{1, 2\}$.

(6) If ϕ is $(\psi \leftrightarrow \theta)$, then $V^*(\phi) = (V^*(\psi) \cap V^*(\theta)) \cup (V^*(\sim\psi) \cap V^*(\sim\theta))$.

(7) If ϕ is $\Box\psi$, then $V^*(\phi) = W$ iff $V^*(\psi) = W$, otherwise $V^*(\phi) = \emptyset$.

(8) If ϕ is $\Diamond\psi$, then $V^*(\phi) = W$ iff $V^*(\psi) \neq \emptyset$, otherwise $V^*(\phi) = \emptyset$.

For a given model $\mathfrak{m}$, we call this function V^* the *valuation function* of $\mathfrak{m}$. From here there is a straightforward way to define truth in a world in a model:

Definition 9.7. A sentence ϕ of MSL is *true in world* w in model $\mathfrak{m} = \langle w_0, W, V\rangle$ where $w \in W$ iff $w \in V^*(\phi)$.

Although there isn't as much intuitive significance to it, we can define truth in a model:

Definition 9.8. A sentence ϕ of MSL is *true* in model $\mathfrak{m} = \langle w_0, W, V\rangle$ iff $V^*(\phi) = W$.

Perhaps it's best not to use "truth" here, but instead something like "truth everywhere"; so we might say that ϕ is true everywhere in $\mathfrak{m}$ iff $V^*(\phi) = W$, for V^* the valuation function of $\mathfrak{m}$. The significant concepts are (1) truth in a world in a model (def. 9.7), and (2) truth at all worlds in a model (def. 9.8).

The concepts of logical truth, logical falsity, and logical indeterminacy can be adapted to MSL as well:

Definition 9.9. ϕ is a *modal truth* iff for all models $\mathfrak{m} = \langle w_0, W, V\rangle$, $V^*(\phi) = W$.

Definition 9.10. ϕ is a *modal falsehood* iff for all models $\mathfrak{m} = \langle w_0, W, V\rangle$, $V^*(\phi) = \emptyset$.

Definition 9.11. ϕ is *modally contingent* iff there's a model $\mathfrak{m} = \langle w_0, W, V\rangle$ such that $V^*(\phi) \neq \emptyset$ and there's a model $\mathfrak{m} = \langle w_0, W, V\rangle$ such that $V^*(\phi) \neq W$. (Note: It can be the same model for both.)

Equivalently put, a sentence ϕ is a modal truth iff it's true everywhere on all models $\mathfrak{m}$; i.e. iff for every world w in every model $\mathfrak{m}$, ϕ is true in w in $\mathfrak{m}$. Similarly, we can alternatively say that a sentence ϕ is a modal falsity iff for every world w in every model $\mathfrak{m}$, ϕ is false in w in $\mathfrak{m}$.

Finally, we also can define when a set Δ of MSL sentences entails a sentence ϕ.

Definition 9.12. A sentence ϕ is *entailed* by a set of sentence Δ iff, for every model $\mathfrak{m} = \langle w_0, W, V\rangle$ and world $w \in W$, whenever every sentence in Δ is true in w in $\mathfrak{m}$, then ϕ is also true in w in $\mathfrak{m}$; i.e. iff, for every model $\mathfrak{m} = \langle w_0, W, V\rangle$ and world $w \in W$, $\bigcap_{\psi\in\Delta} V^*(\psi) \subseteq V^*(\phi)$.

Example 9.13. $\Diamond A \rightarrow \Box\Diamond A$ is a modal truth.

Proof: Note that $w \in V^*(\sim\psi) \cup V^*(\theta)$, for V^* the valuation function of some model $\mathfrak{m}$ and w a world in the set W of $\mathfrak{m}$, iff if $w \in V^*(\psi)$, then $w \in V^*(\theta)$. So, suppose $\Diamond A$ is true at some w in some model $\mathfrak{m}$, i.e. suppose that $w \in V^*(\Diamond A)$ for V^* the valuation function of $\mathfrak{m}$. By (8) of definition 9.6, either $V^*(\Diamond A) = W$ or $V^*(\Diamond A) = \emptyset$.

Since $w \in V^*(\Diamond A)$, $V^*(\Diamond A) \neq \emptyset$. So, $V^*(\Diamond A) = W$. Now Consider $\Box\Diamond A$. Since $V^*(\Diamond A) = W$, by (7) of definition 9.6 it follows that $V^*(\Box\Diamond A) = W$. So, $\Box\Diamond A$ is true at every element of W, including w. Therefore, $w \in V^*(\Box\Diamond A)$. Since w was arbitrary, this means that for $\mathfrak{m}$, $V^*(\sim\psi) \cup V^*(\theta) = W$. Since $\mathfrak{m}$ was arbitrary, this means that $V^*(\sim\psi) \cup V^*(\theta) = W$ for all $\mathfrak{m}$. ■

Example 9.14. There are sentences ϕ and ψ for which $\Box(\phi \vee \psi) \rightarrow (\Box\phi \vee \Box\psi)$ is not a modal truth.

Proof: Consider a model $\mathfrak{m} = \langle w_0, W, V\rangle$ with the set $W = \{w_0, w_1\}$. Let $\phi = B$ and let $\psi = \ \sim B$. Let $V(B) = \{w_0\}$. By (2) in definition 9.6, $V^*(\sim B) = \{w_1\}$. By (4) in definition 9.6, $V^*(B \vee \sim B) = \{w_0, w_1\}$; hence by (7) of definition 9.6 $V^*(\Box(B \vee \sim B)) = \{w_0, w_1\}$. Thus, $\Box(B \vee \sim B)$ is true at each w in W, including world w_0. Consider then $(\Box B \vee \Box\sim B)$. By (7) of definition 9.6, $V^*(\Box B) = \emptyset$ since $V^*(B) = \{w_0\} \neq W$. Again, $V^*(\Box\sim B) = \emptyset$ since $V^*(\sim B) = \{w_1\} \neq W$. So, by (4) in definition 9.6, $V^*(\Box B \vee \Box\sim B) = \emptyset$. Thus, $(\Box B \vee \Box\sim B)$ is false at each w in W, including world w_0. Therefore, $\Box(B \vee \sim B) \rightarrow (\Box B \vee \Box\sim B)$ is not true in w_1 in $\mathfrak{m}$; so it is not a modal truth. ■

You should see a strong similarity here with the fact that $\forall x(Bx \vee \sim Bx) \rightarrow (\forall x Bx \vee \forall x\sim Bx)$ is not a logical truth.

Example 9.15. All of the following are modal truths, for any substitution of sentences ϕ and ψ. We leave the proofs as exercises to the reader.

(1) $\Box\Box\phi \rightarrow \Box\phi$
(2) $\Box\Diamond\phi \rightarrow \Diamond\phi$
(3) $\Diamond\phi \rightarrow \Diamond\Diamond\phi$
(4) $\Box\phi \rightarrow \Diamond\Box\phi$
(5) $\sim\Diamond\phi \leftrightarrow \Box\sim\phi$
(6) $\Diamond\sim\phi \leftrightarrow \sim\Box\phi$
(7) $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$
(8) $(\Box\phi \vee \Box\psi) \rightarrow \Box(\phi \vee \psi)$
(9) $(\Diamond\phi \vee \Diamond\psi) \leftrightarrow \Diamond(\phi \vee \psi)$
(10) $\Box(\phi \wedge \psi) \leftrightarrow (\Box\phi \wedge \Box\psi)$
(11) $\Diamond(\phi \wedge \psi) \rightarrow (\Diamond\phi \wedge \Diamond\psi)$
(12) $\sim\Diamond\sim\phi \leftrightarrow \Box\phi$
(13) $\Box(\phi \rightarrow \psi) \rightarrow (\Diamond\phi \rightarrow \Diamond\psi)$
(14) $\Diamond(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Diamond\psi)$
(15) $\sim\Diamond\phi \rightarrow \Box(\phi \rightarrow \psi)$
(16) $\Box(\phi \leftrightarrow \psi) \rightarrow (\Box\phi \leftrightarrow \Box\psi)$

One of the most distinctive features of the models we're using here (def. 9.5) and the definitions of truth (def. 9.6) and modal truth (def. 9.9) that come with them is that only the innermost modality of a string of modalities matters. This follows from the iteration of the following basic facts.

Theorem 9.16. The following are modal truths, for every sentence ϕ.

(1) $\Box\Box\phi \leftrightarrow \Box\phi$
(2) $\Box\Diamond\phi \leftrightarrow \Diamond\phi$
(3) $\Diamond\phi \leftrightarrow \Diamond\Diamond\phi$
(4) $\Box\phi \leftrightarrow \Diamond\Box\phi$

Again we leave the rigorous proofs to the reader. But, from the intuitive semantic point of view this is because the modalities $\Box$ and $\Diamond$ behave like quantifiers over a fixed domain with only one variable that they can quantify. Thus any quantification after the first is vacuous. There are other ways of defining modal models of in which (roughly) the domain of quantification over worlds isn't always the same. The schemas in theorem 9.16 don't represent modal truths in these modal logics. We do not discuss alternative modal model definitions in this text.

9.2.3 Derivations in $S5$

We would like a derivation system which is sound and complete for the semantics just defined in the last section, i.e. so that a sentence of MSL is a modal truth (def. 9.9) iff it's derivable in that system. We get such a system, usually called $S5$, by adding introduction and elimination rules for $\Box$ and $\Diamond$ to SD.

Name	**Given**	**May Add**
$\Box$-*Elim*	$\Box\phi$	ϕ
$\Box$-*Intro*	ϕ (*)	$\Box\phi$
$\Diamond$-*Elim*	$\Diamond\phi$, $\phi\rightarrow\psi$ (*), (**)	ψ
$\Diamond$-*Intro*	ϕ	$\Diamond\phi$

Table 9.1: Basic Rules of $S5$

There are two restrictions to the rules: (*) in $\Box$-*Intro* and $\Diamond$-*Elim*, the rule can only be applied if all the open assumptions have modal prefixes. (**) In $\Diamond$-*Elim*, the rule can only be applied if ψ has a modal prefix.

Definition 9.17. A sentence ϕ has a *modal prefix* iff

(1) it begins with $\Box$;
(2) it begins with $\Diamond$; or
(3) it is of the form $\sim\phi$, $\sim\sim\phi$, ... where ϕ is type (1) or (2).

Example 9.18. As they are all modal truths, every instance of the schemas in example 9.15 (on the previous page) and theorem 9.16 (on the previous page) can be derived in $S5$.

Just as before we can add shortcut rules to $S5$. These we call *modal negation* rules, similar to the quantification negation rules (table 7.19, on page 170). It can be shown that anything we can prove using $S5$ and the following shortcut rules can be proved in $S5$ alone.

Name	Given	May Add
MN	$\sim\Box\phi$	$\Diamond\sim\phi$
	$\sim\Diamond\phi$	$\Box\sim\phi$
	$\sim\Diamond\sim\phi$	$\Box\phi$
	$\sim\Box\sim\phi$	$\Diamond\phi$

Table 9.2: Modal Negation Shortcut Rules

Proving soundness for $S5$ is very similar to proving it for SD. In fact, we start with the proof for SD and add four more parts: one establishing the soundness of each new modal rule. Proving completeness for $S5$ requires a more complicated modification of the methods for SD. For logically valid sentences we still need only provide a derivation, but for invalid sentences we must construct a model with various worlds and a model falsifying the sentence.

9.2.4 History of Modal Logic

It's worth discussing some historical background. (Part of this overview is drawn from Roberta Ballarin's SEP entry (2010); the reader should also consult Goldblatt 2006.) Just as with sentential and quantificational logic, work on modal logic started as work on deduction systems without any semantics, and started with sentential logic without quantification. C.I. Lewis is well known for his early work in the 1910's on derivation systems for sentential modal logic. In his (1932), coauthored with Cooper Langford, Lewis lays out his famous systems $S1$–$S5$. ($S5$, still of much interest today, is the system we focus on here.) It wasn't until Ruth Barcan Marcus (1946b; 1946a; 1947) and to a lesser extent Rudolf Carnap (1946; 1947) that systems were developed for quantificational modal logic (see also Marcus's (1993)).

While Marcus and Carnap did work on semantics for quantified modal logic (with Carnap drawing on Leibniz's conception of a "possible world"), it wasn't until the work of Stig Kanger (1957), Richard Montague (1960), Jaakko Hintikka (1961), A.N. Prior (1957), and (especially) Saul Kripke (1959; 1963a; 1963b; 1965) that modern looking semantics (for both sentential and quantificational modal logic) were devised.[6] These are often called relational, or Kripke, semantics and are what we'll look at here (at

[6]This claim isn't quite true: matrix, or algebraic, semantics for modal logics have a long history, and they surely count as "modern looking".

first in simplified form), although only for sentential modal logic. Like quantificational logic, there are alternative semantics. Richard Montague (1970) and Dana Scott (1970) independently developed what's typically called neighborhood semantics (Arló-Costa and Pacuit 2006), while matrix (or algebraic) semantics have a long history that goes back before the development of relational semantics (see Cocchiarella and Freund 2008: ch. 3 for a textbook treatment). We do not bring these up at all, instead focusing on the basics of relational semantics.

For those readers looking to pursue modal logic further, we recommend the following three textbooks (listed from most basic to most advanced): (Beall and van Fraassen 2003), (Hughes and Cresswell 1996), and (Cocchiarella and Freund 2008). For handbook-style treatments, see (Cresswell 2001), (Bull and Segerberg 2001), (Zakharyaschev et al. 2001), and (Garson 2001).

9.3 Quantifier Logic with Identity

9.3.1 Introduction

We aren't able to say in QL that one object is identical to another, which is often an important fact. We address this limitation of SL by extending the language.

Before we give this extension, it's worth mentioning a few motivations. First, say we have two constants t and s. It turns out that there's no sentence ϕ of QL (or even set of sentences of QL) which is true on all and only the models $\mathfrak{m}$ where $\mathfrak{m}(t) = \mathfrak{m}(s)$. If we translate English names as constants in QL, then translations of the following sentences

52. Cicero is Tully.
53. Clark Kent is Superman.
54. Mark Twain is Samuel Clemens.

allow for models where (say) Cicero is not Tully, or Clark Kent is not Superman. Perhaps more importantly, this means that there's no such translation of 54, along with a translation of

55. Mark Twain wrote *The Adventures of Tom Sawyer*.

which entails a translation of

56. Samuel Clemens wrote *The Adventures of Tom Sawyer*.

We can get around this difficulty if we instead translate names in English as predicates of QL that have only one member. This is a somewhat unintuitive work-around, however.

Next, say we want a sentence ϕ (or set of sentences Δ) which contains a predicate P and which is satisfied by all and only models $\mathfrak{m}$ such that $\mathfrak{m}(P)$ contains only a single element. We leave it to readers to convince themselves that there are no sentences

like this in QL. The point generalizes to any size set: for any number n, there is no sentence (or set of sentences) of QL which is satisfied by all and only models $\mathfrak{m}$ such that $\mathfrak{m}(P)$ contains exactly n elements. (This holds when n is infinite too.) But, if we extend QL to capture identity claims, then we are able to find such sentences.

Thinking about translations of English sentences again, this inability to capture cardinality claims suggests that QL can't provide satisfactory translations for English sentences like:

57. There exists exactly one bad apple.
58. At least five different philosophers tried to change that light bulb.
59. There exists an infinite number of things.

So by extending QL to capture identity claims we'll also be able to capture cardinality claims, and translate sentences like these.

9.3.2 The Language QLI

To get the language QLI, we add to the basic symbols of QL (def. 4.1, on page 84) the identity symbol '=' and add a new base clause to the recursive definition of a QL formula (def. 4.2, on page 84).

Definition 9.19. The *formulas of QLI* are given by the following recursive definition:

Base Clauses:

(1) A sentence letter (atomic sentence of SL) is an atomic formula.
(2) An n-place predicate followed by n occurrences (tokens) of individual constants or variables is an atomic formula.
(3) Given two individual terms t and s (see Sec. 4.2.2), $t = s$ is an atomic formula.

So $b = d$, $b = x$, $y = d$, and $x = x$ are all examples of atomic formulas.

Generating Clauses:

(1) If ϕ is a formula, then so is $\sim\phi$.
(2) If ϕ and θ are formulas, then so are $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.
(3) If all of $\phi_1, \phi_2, \phi_3, \phi_4, \ldots, \phi_n$ are formulas (the list must include at least two formulas and be finite), then so are $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \ldots \vee \phi_n)$.
(4) If ϕ is a formula and it does not contain an expression of the form $\forall\alpha$ or $\exists\alpha$ for some QL variable α, then $\forall\alpha\phi$ and $\exists\alpha\phi$ are formulas.

Closure Clause: A string of symbols is a formula iff it can be generated by the clauses above.

Note that this definition of formulas of QLI is exactly the same as that for formulas of QL (def. 4.2, on page 84), except for the new third base clause.

9.3.3 Truth, Logical Truth, and Entailment in GQLI

The models for QLI are just the models for QL (recall def. 4.5, on page 86), but obviously we have to extend the definition of truth to account for sentences with the identity symbol, '='.

Definition 9.20. The following clauses fix when a sentence of QLI is *true* (or *false*) on a model $\mathfrak{m}$:

(1) A sentence letter ϕ is true on $\mathfrak{m}$ iff $\mathfrak{m}$ assigns true to it, i.e. iff $\mathfrak{m}(\phi) = \mathsf{T}$.

(2) An atomic sentence Pt with a 1-place predicate P and an individual term t is true on $\mathfrak{m}$ iff what $\mathfrak{m}$ assigns to the individual term t is in the set $\mathfrak{m}$ assigns to the predicate, i.e. iff $\mathfrak{m}(t) \in \mathfrak{m}(P)$.

(3) An atomic sentence $Pt_1 \dots t_n$ with an n-place predicate P is true on $\mathfrak{m}$ iff $\langle \mathfrak{m}(t_1), \mathfrak{m}(t_2), \dots, \mathfrak{m}(t_n) \rangle \in \mathfrak{m}(P)$.

(4) An atomic sentence $t = s$ is true on $\mathfrak{m}$ iff $\mathfrak{m}(t) = \mathfrak{m}(s)$.

(5) A negation $\sim\phi$ is true on $\mathfrak{m}$ iff the unnegated formula ϕ is false on $\mathfrak{m}$.

(6) A conjunction $(\phi_1 \wedge \dots \wedge \phi_n)$ is true on $\mathfrak{m}$ iff all conjuncts $\phi_1, \dots, \phi_n$ are true on $\mathfrak{m}$.

(7) A disjunction $(\phi_1 \vee \dots \vee \phi_n)$ is true on $\mathfrak{m}$ iff at least one disjunct $\phi_1, \dots, \phi_n$ is true on $\mathfrak{m}$.

(8) A conditional $(\psi \rightarrow \phi)$ is true on $\mathfrak{m}$ iff the LHS ψ is false or the RHS ϕ is true on $\mathfrak{m}$.

(9) A biconditional $(\psi \leftrightarrow \phi)$ is true on $\mathfrak{m}$ iff both sides, ψ and ϕ, have the same truth value on $\mathfrak{m}$.

(10) A universal quantification $\forall\alpha\phi$ is true on $\mathfrak{m}$ iff $\phi t/\alpha$ is true on *all* t-variants of $\mathfrak{m}$ (where t is the first *constant* not contained in ϕ).

(11) An existential quantification $\exists\alpha\phi$ is true on $\mathfrak{m}$ iff $\phi t/\alpha$ is true on *some* t-variant of $\mathfrak{m}$ (where t is the first *constant* not contained in ϕ).

(12) A sentence ϕ is false on $\mathfrak{m}$ iff ϕ is not true on $\mathfrak{m}$.

This definition is exactly the same as the one given for QL, except for the added clause (4) which covers the new atomic sentence added in definition 9.19.

The definitions of quantificational truth (def. 3.37), falsehood (def. 3.38), and contingency (def. 3.39) are the same as they were for QL. The definition of entailment is also the same, and likewise for the other relations defined in section 4.2.4.

Example 9.21. Consider any model $\mathfrak{m}$ which includes Cicero and Tully in its universe and assigns c to Cicero and d to Tully. (Cicero was a great Roman orator from the first century BC, and 'Tully' is an anglicized version of his name.) Then there exists a sentence ϕ of QLI such that all and only those models $\mathfrak{m}$ in which Cicero is Tully

make ϕ true. One such sentence is $\mathrm{c} = \mathrm{d}$. Hence, $\mathrm{c} = \mathrm{d}$ is a reasonable translation of the English sentence 'Cicero is Tully.'

Example 9.22. Consider the sentence $(\mathrm{c} = \mathrm{d} \wedge R\mathrm{c}) \rightarrow R\mathrm{d}$ and consider just those models $\mathfrak{m}$ which include Cicero and Tully in the universe, assign c to Cicero and d to Tully, and assign to R the set of all Romans. Then any of those models which make the sentence true are such that either $\mathfrak{m}(\mathrm{d}) \in \mathfrak{m}(R)$, or either $\mathfrak{m}(\mathrm{c}) \neq \mathfrak{m}(\mathrm{d})$ or $\mathfrak{m}(\mathrm{c}) \notin \mathfrak{m}(R)$. Hence the sentence is a reasonable translation of the English sentence 'If Cicero is Tully and Cicero is Roman, then Tully is Roman.'

Example 9.23. Consider the sentence $\exists x(Ax \wedge \forall y(Ay \rightarrow y = x))$. A model $\mathfrak{m}$ makes the sentence true iff $\mathfrak{m}(A)$ contains exactly one element. Hence the sentence is a reasonable translation of the English sentence 'There exists exactly one apple', or any other English sentence that differs from this one only in a change of the predicate 'apple'.

Example 9.24. Consider the sentence

$$\exists w \exists z(Aw \wedge Az \wedge w \neq z \wedge \forall v(Av \rightarrow (v = w \vee v = z))).$$

The two existential quantifiers with the negative identity guarantee that there are at least two instances of A, and the universal quantifier guarantees that these are the only ones. So a model $\mathfrak{m}$ makes the sentence true iff $\mathfrak{m}(A)$ contains exactly two elements. Hence the sentence is a reasonable translation of the English sentence 'There exists (exactly) two apples.'

Example 9.25. Consider the sentence

$$\exists x(Ax \wedge \forall y(Ay \rightarrow y = x)) \rightarrow \\ \sim \exists w \exists z(Aw \wedge Az \wedge w \neq z \wedge \forall v(Av \rightarrow (v = w \vee v = z))).$$

This sentence is a reasonable translation of the English sentence 'If there exists exactly one apple, then there doesn't exist exactly two apples.'

Example 9.26. Consider the sentence $C\mathrm{ba} \wedge N\mathrm{b} \wedge \forall x((C x\mathrm{a} \wedge Nx) \rightarrow x = \mathrm{b})$. This sentence is true in those models $\mathfrak{m}$ where only one element of the domain is both in the set $\mathfrak{m}(N)$ and stands in relation C to $\mathfrak{m}(\mathrm{a})$. So, if $C\mathrm{ba}$ means "b is a child of a", $N\mathrm{b}$ means "b is male", a is Arnold, and b is Bob, this sentence would make a reasonable translation of the English sentence 'Bob is Arnold's only son.'

Example 9.27. Consider the sentence $\exists y_1 \exists y_2 \forall x(x = y_1 \vee x = y_2)$. It is true in all and only those models $\mathfrak{m}$ that have two or less elements in their universe. So, the sentence would make a reasonable translation of the English sentence 'At most only two things exist.'

Example 9.28. Consider the sentence $\exists x \exists y x \neq y$. It is true in all and only those models $\mathfrak{m}$ with two or more elements in their domain. So, it would make a reasonable translation of the English sentence 'At least two things exist.'

Example 9.29. Consider the sentence $\exists x \exists y \exists z (x \neq y \wedge x \neq z \wedge y \neq z)$. It's true in all and only those models $\mathfrak{m}$ with three or more elements in their domain. So, it would make a reasonable translation of the English sentence 'At least three things exist.'

Example 9.30. Consider the sentence

$$\begin{aligned}\forall x \{ Gx \rightarrow \forall y [(Py \wedge Ryx) \rightarrow \exists z_1 \exists z_2 \exists z_3 \\ (Iz_1 \wedge Iz_2 \wedge Iz_3 \wedge Gxz_1y \wedge Gxz_2y \wedge Gxz_3y \wedge \\ z_1 \neq z_2 \wedge z_1 \neq z_3 \wedge z_2 \neq z_3)] \}.\end{aligned}$$

It would make a reasonable translation of the English sentence 'Every genie grants anyone who releases them at least three wishes.'

Example 9.31. Even with identity, there's no one sentence of GQLI which is true in all and only those models with infinite domains.[7] But, there's an infinite set of sentences Δ such that a model makes every sentence of Δ true iff it has an infinite domain. Define:

$$\begin{aligned}\bigwedge_{i \neq j}^{n} x_i \neq x_j \quad \Leftrightarrow \quad & (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge \ldots \wedge x_1 \neq x_n \wedge \\ & x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_2 \neq x_5 \wedge \ldots \wedge x_2 \neq x_n \wedge \\ & x_3 \neq x_4 \wedge x_3 \neq x_5 \wedge x_3 \neq x_6 \wedge \ldots \wedge x_3 \neq x_n \wedge \\ & \vdots \\ & x_{n-2} \neq x_{n-1} \wedge x_{n-2} \neq x_n \wedge \\ & x_{n-1} \neq x_n)\end{aligned}$$

Then a model makes every sentence in the following list true iff its universe is infinite:

$$\exists x_1 \exists x_2 \bigwedge_{i \neq j}^{2} x_i \neq x_j$$

$$\exists x_1 \exists x_2 \exists x_3 \bigwedge_{i \neq j}^{3} x_i \neq x_j$$

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \bigwedge_{i \neq j}^{4} x_i \neq x_j$$

[7] There are some single sentences of GQLI which are such that the only models that make them true have infinite domains, but not every model with an infinite domain makes these true. So, these sentences can't really be thought of as saying that there exists an infinite number of things. They say more than that.

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 \bigwedge_{i \neq j}^{5} x_i \neq x_j$$

$$\vdots$$

We leave it to the reader to show that this is true.

9.3.4 Derivations in GQDI

We get a sound and complete derivation system for QDI by adding an introduction and elimination rule for identity.

Name	Given	May Add
$=$-*Intro*		$t = t$
$=$-*Elim*	ϕ, $t = s$	$\phi t/s$

Table 9.3: Basic Rules of QDI

Note that since only sentences can appear on lines of derivations, t and s must be constants. Also, note that $=$-*Intro* is like *Assume* insofar as neither is "applied" to previous lines; both allow you to write a sentence that fits the may-add schema at any point in a derivation.

Like QD, QDI is both sound and strongly complete.

Theorem 9.32. QDI Soundness Theorem: For all sentences ϕ in QLI and sets of sentences Δ, if $\Delta \vdash \phi$ in QDI, then $\Delta \models \phi$.

Theorem 9.33. QDI Strong Completeness Theorem: For all sentences ϕ in QLI and sets of sentences Δ, if $\Delta \models \phi$ in QDI, then $\Delta \vdash \phi$.

We won't prove either the soundness or completeness of QDI, but a few remarks are in order. There is nothing essentially different about the soundness proof of QDI, compared with the proof for QD. We can still use the general approach from section 8.4 (on page 192) to prove the (strong) completeness of QDI, but there are two important differences. Here we briefly mention where changes need to be made without describing how those changes can be made.

The first difference concerns our search for contradictions (recall Step 4 from Sec. 8.4.3, on page 195). It's not enough to take the conjunction of all the matrix instances (by $\wedge$-*Intro*), use *Distribution* to get the conjunction into DNF and check whether every disjunct contains a contradiction. The reason is that there might be some disjuncts that don't contain a contradiction, but do contain an ion ϕ, an ion $\sim\psi$ where $\phi = \phi t/s$, and the ion $t = s$. For example, consider the disjunct $G\mathrm{a} \wedge \sim G\mathrm{b} \wedge \mathrm{a} = \mathrm{b}$. A

contradiction can be derived from these disjuncts using =-*Elim*, so we need to adjust our procedure accordingly.

The second difference concerns how we construct the model when The Method (or The Strong Method) doesn't produce a contradiction. The old procedure works by assigning distinct elements to each constant that appears in the list of sentences produced by The Method. So, if we could construct a model that made all the sentences produced by The Method (when it doesn't end in a contradiction) true by the old procedure, that model would assign distinct elements to each constant. But there are sets Δ of QLI sentences that both contain ions of the form $t = s$ and are consistent. Clearly no model that assigns distinct elements to the constants t and s could make all the sentences in such a set Δ true.

Our procedure is to take the model generated by the previous method and modify it by systematically changing the constant assignments so that if s = t is in the Master Matrix list s and t are both assigned the same constant.

9.4 Exercises

9.4.1 TFT in Many-Valued Logic

Each of the following sentences are 2-valued TFT. Which are also 3-valued TFT? Which of the 3-valued TFT are n-valued TFT for all finite n?

1. $\sim\sim A \leftrightarrow A$
2. $(A \rightarrow B) \rightarrow (\sim A \vee B)$
3. $(\sim A \vee B) \rightarrow (A \rightarrow B)$
4. $\sim(A \vee B) \leftrightarrow (\sim A \wedge \sim B)$
5. $A \rightarrow (\sim A \rightarrow B)$
6. $A \rightarrow ((A \rightarrow B) \rightarrow B)$
7. $(A \wedge \sim A) \rightarrow B$
8. $(A \rightarrow (B \vee C)) \rightarrow ((A \rightarrow B) \vee C)$
9. $(A \leftrightarrow B) \vee (A \leftrightarrow C) \vee (A \leftrightarrow D) \vee (B \leftrightarrow C) \vee (B \leftrightarrow D) \vee (C \leftrightarrow D)$
10. $((A \vee B) \wedge \sim A) \rightarrow B$

9.4.2 Modal Truths in MSL #1

For each schema below, show that it's a modal truth for all MSL sentences ϕ and ψ. These are from example 9.15 (on page 222).

1. $\Box\Box\phi \rightarrow \Box\phi$
2. $\Box\Diamond\phi \rightarrow \Diamond\phi$
3. $\Diamond\phi \rightarrow \Diamond\Diamond\phi$
4. $\Box\phi \rightarrow \Diamond\Box\phi$
5. $\sim\Diamond\phi \leftrightarrow \Box\sim\phi$
6. $\Diamond\sim\phi \leftrightarrow \sim\Box\phi$
7. $\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$
8. $(\Box\phi \vee \Box\psi) \rightarrow \Box(\phi \vee \psi)$

9. $(\Diamond \phi \vee \Diamond \psi) \leftrightarrow \Diamond (\phi \vee \psi)$
10. $\Box (\phi \wedge \psi) \leftrightarrow (\Box \phi \wedge \Box \psi)$
11. $\Diamond (\phi \wedge \psi) \rightarrow (\Diamond \phi \wedge \Diamond \psi)$
12. $\sim \Diamond \sim \phi \leftrightarrow \Box \phi$
13. $\Box (\phi \rightarrow \psi) \rightarrow (\Diamond \phi \rightarrow \Diamond \psi)$
14. $\Diamond (\phi \rightarrow \psi) \rightarrow (\Box \phi \rightarrow \Diamond \psi)$
15. $\sim \Diamond \phi \rightarrow \Box (\phi \rightarrow \psi)$
16. $\Box (\phi \leftrightarrow \psi) \rightarrow (\Box \phi \leftrightarrow \Box \psi)$

9.4.3 Modal Truths in MSL #2

For each schema below, show that it's a modal truth for all MSL sentences ϕ. These are from theorem 9.16, on page 222.

1. $\Box \Box \phi \leftrightarrow \Box \phi$
2. $\Box \Diamond \phi \leftrightarrow \Diamond \phi$
3. $\Diamond \phi \leftrightarrow \Diamond \Diamond \phi$
4. $\Box \phi \leftrightarrow \Diamond \Box \phi$

9.4.4 Modal Truths in MSL #3

Show whether each of the following is a modal truth.

1. $\vdash \Diamond A \leftrightarrow \sim \Box \sim A$
2. $\vdash \Box A \rightarrow \Box \Box A$
3. $\vdash \Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
4. $\vdash \Diamond (A \vee B) \leftrightarrow (\Diamond A \vee \Diamond B)$
5. $\vdash (\Diamond A \wedge \Diamond B) \rightarrow \Diamond (A \wedge B)$
6. $\vdash \Box (A \rightarrow B) \rightarrow (A \rightarrow \Box B)$
7. $\vdash A \rightarrow \Box A$
8. $\vdash \Diamond (A \rightarrow B) \rightarrow (\Box A \rightarrow \Diamond B)$
9. $\vdash \Box (A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B)$
10. $\vdash \Box (A \wedge B) \leftrightarrow (\Box A \wedge \Box B)$
11. $\vdash \Box \Diamond A \rightarrow \Diamond \Box A$

9.4.5 Derivations in MSL

Write derivations for each schema below in $S5$. Try finding derivations both with and without the modal negation rules. Again, these are from example 9.15 (on page 222).

1. $\Box \Box \phi \rightarrow \Box \phi$
2. $\Box \Diamond \phi \rightarrow \Diamond \phi$
3. $\Diamond \phi \rightarrow \Diamond \Diamond \phi$
4. $\Box \phi \rightarrow \Diamond \Box \phi$
5. $\sim \Diamond \phi \leftrightarrow \Box \sim \phi$
6. $\Diamond \sim \phi \leftrightarrow \sim \Box \phi$
7. $\Box (\phi \rightarrow \psi) \rightarrow (\Box \phi \rightarrow \Box \psi)$
8. $(\Box \phi \vee \Box \psi) \rightarrow \Box (\phi \vee \psi)$
9. $(\Diamond \phi \vee \Diamond \psi) \leftrightarrow \Diamond (\phi \vee \psi)$
10. $\Box (\phi \wedge \psi) \leftrightarrow (\Box \phi \wedge \Box \psi)$
11. $\Diamond (\phi \wedge \psi) \rightarrow (\Diamond \phi \wedge \Diamond \psi)$
12. $\sim \Diamond \sim \phi \leftrightarrow \Box \phi$

13. $\Box(\phi \to \psi) \to (\Diamond \phi \to \Diamond \psi)$
14. $\Diamond(\phi \to \psi) \to (\Box \phi \to \Diamond \psi)$
15. $\sim\Diamond \phi \to \Box(\phi \to \psi)$
16. $\Box(\phi \leftrightarrow \psi) \to (\Box \phi \leftrightarrow \Box \psi)$

9.4.6 Derivations in QLI

Write derivations for each of the following in QDI.

1. $\vdash \forall x x = x$
2. $\vdash \forall x \forall y (x = y \to y = x)$
3. $\vdash \forall x \forall y \forall z ((x = y \wedge y = z) \to x = z)$
4. $\vdash \forall x (Gx \leftrightarrow \exists y (x = y \wedge Gy))$
5. $\vdash \forall x (Gx \leftrightarrow \forall y (x = y \to Gy))$
6. $\vdash \forall x \forall y (x = y \to (Gx \leftrightarrow Gy))$
7. $\vdash \exists x \forall y (Gy \leftrightarrow y = x) \to (\exists x Gx \wedge \forall x \forall y ((Gx \wedge Gy) \to x = y))$
8. $\vdash (\exists x Gx \wedge \forall x \forall y ((Gx \wedge Gy) \to x = y)) \to \exists x \forall y (Gy \leftrightarrow y = x)$
9. $\vdash \forall x \exists y (y \neq x \wedge Gy) \to \exists x \exists y (x \neq y \wedge (Gx \wedge Gy))$
10. $\vdash \exists x \exists y (x \neq y \wedge (Gx \wedge Gy)) \to \forall x \exists y (y \neq x \wedge Gy)$
11. $\vdash (\forall x \exists y Gxy \wedge \forall x \sim Gxx) \to \forall x \exists y (x \neq y \wedge Gxy)$
12. $\vdash (G\mathrm{a} \wedge \sim G\mathrm{b}) \to \exists x \exists y x \neq y$
13. $\vdash (G\mathrm{a} \wedge \forall x (x \neq \mathrm{a} \to Gx)) \leftrightarrow \forall x Gx$
14. $\vdash \forall x (x \neq \mathrm{a} \to Gx) \to \forall x \forall y (x \neq y \to (Gx \vee Gy))$
15. $\vdash \exists x \forall y (y \neq x \to Gy) \to \forall x \forall y (x \neq y \to (Gx \vee Gy))$
16. $\vdash \forall x \forall y (x \neq y \to (Gx \vee Gy)) \to \exists x \forall y (y \neq x \to Gy)$
17. $\vdash \exists y \forall x x = y \to (\forall x Gx \vee \forall x \sim Gx)$
18. $\vdash \forall x \forall y \forall z (x = y \vee x = z \vee y = z) \to (\forall x Gx \vee \forall x (Gx \to Hx) \vee \forall x (Gx \to \sim Hx))$

9.4.7 Translations into QLI

Translate each of the following English sentences into QLI sentences about the model $\mathfrak{m}$ given in table 9.4 (on the next page).

1. Bob is Arnold's only son.
2. Arnold has at least two children.
3. Arnold has at least two sons.
4. Arnold has at most two sons.
5. Arnold has exactly two sons.
6. Bob is Carol's only brother.
7. Bob is an only child.
8. Bob is an only son.
9. Bob has at least two sisters.
10. Bob has just one sister.
11. Everyone has a sister.
12. Carol's mother is Bob's only sister.
13. Bob has at least two grandchildren.
14. Diane only dates Bob.
15. Bob only dates daughters.
16. Bob dates only daughters.
17. Do (16) another way.
18. Bob only dates only daughters.
19. Bob dates only only daughters.
20. Only Bob only dates only daughters.

	Symbol	**Assignment**
Universe:		All people
Constants:	a	Arnold
	b	Bob
	c	Carol
	d	Diane
1 place predicates:	M'	Male
	E'	Female
2 place predicates:	P''	is the parent of
	D''	dates

Table 9.4: Interpretation for Translations in Section 9.4.7

Appendix A: List of Derivation Rules

Derivation Systems:

SD	Set 1
SD$^+$	Sets 1,2,3
QD	Sets 1,4
QD$^+$	Sets 1,2,3,4,5
QD^+_m	Sets 1,2,3,4,5,6,7,8
$S5$	Sets 1,9
$S5^+$	Sets 1,2,3,9,10
QDI	Sets 1,4,11
QDI$^+$	Sets 1,2,3,4,5,11

Name	**Given**	**May Add**
Set 1: Basic Rules of GSD, table 6.2 (on page 125)		
Ass.		$\mid \phi$
Rep.	ϕ	ϕ
$\rightarrow$*-Elim*	$\theta \rightarrow \psi$, θ	ψ
$\rightarrow$*-Intro*	$\mid \theta \Rightarrow \psi$	$\theta \rightarrow \psi$, Box $\mid \theta \Rightarrow \psi$
$\wedge$*-Elim*	$\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n$	Conjunction of any proper subset of the conjuncts
$\wedge$*-Intro*	θ_1, θ_2, ... θ_n	$\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n$
$\vee$*-Elim*	$\theta_1 \vee \theta_2 \vee ... \vee \theta_n$, $\theta_1 \rightarrow \psi$, $\theta_2 \rightarrow \psi$,	

Name	Given	May Add
	⋮	
	$\theta_n \rightarrow \psi$	ψ
$\vee$-*Intro*	θ	$\psi_1 \vee \psi_2 \vee ... \vee \psi_n$, where $\theta = \psi_i$ for some i.
$\sim$-*Intro*	$\theta \rightarrow (\psi \wedge \sim\psi)$	$\sim\theta$
$\sim$-*Elim*	$\sim\theta \rightarrow (\psi \wedge \sim\psi)$	θ
$\leftrightarrow$-*Intro*	$\theta \rightarrow \psi$, $\psi \rightarrow \theta$	$\theta \leftrightarrow \psi$
$\leftrightarrow$-*Elim*	$\theta \leftrightarrow \psi$, ψ	θ
$\leftrightarrow$-*Elim*	$\theta \leftrightarrow \psi$, θ	ψ
Set 2: Standard Shortcut Rules of GSD, table 6.39 (on page 145)		
M.T.	$\phi \rightarrow \theta$, $\sim\theta$	$\sim\phi$
D.S.	$\phi_1 \vee ... \vee \phi_i \vee ... \vee \phi_n$, $\sim\phi_i$	$\phi_1 \vee ... \vee \phi_{i-1} \vee \phi_{i+1} \vee ... \vee \phi_n$
	$\phi_1 \vee ... \vee \sim\phi_i \vee ... \vee \phi_n$, ϕ_i	$\phi_1 \vee ... \vee \phi_{i-1} \vee \phi_{i+1} \vee ... \vee \phi_n$
A.C.	$\phi, \sim\phi$	ψ
$\sim/\leftrightarrow$-*Intro*	$\phi \leftrightarrow \psi$	$\sim\phi \leftrightarrow \sim\psi$
Set 3: Exchange Shortcut Rules of GSD, table 6.40 (on page 146)		
DeM	$\sim(\phi_1 \wedge ... \wedge \phi_n)$	$\sim\phi_1 \vee ... \vee \sim\phi_n$
	$\sim\phi_1 \vee ... \vee \sim\phi_n$	$\sim(\phi_1 \wedge ... \wedge \phi_n)$
	$\sim(\phi_1 \vee ... \vee \phi_n)$	$\sim\phi_1 \wedge ... \wedge \sim\phi_n$
	$\sim\phi_1 \wedge ... \wedge \sim\phi_n$	$\sim(\phi_1 \vee ... \vee \phi_n)$
$\sim\sim$-*Elim*	$\sim\sim\phi$	ϕ
$\sim\sim$-*Intro*	ϕ	$\sim\sim\phi$
$\rightarrow/\vee$-*Exchange*	$\phi \rightarrow \theta$	$\sim\phi \vee \theta$
	$\sim\phi \vee \theta$	$\phi \rightarrow \theta$
Contraposition	$\phi \rightarrow \theta$	$\sim\theta \rightarrow \sim\phi$
	$\sim\theta \rightarrow \sim\phi$	$\phi \rightarrow \theta$

Name	Given	May Add
$\sim/\rightarrow$-*Exchange*	$\sim(\phi\rightarrow\theta)$	$\phi\wedge\sim\theta$
	$\phi\wedge\sim\theta$	$\sim(\phi\rightarrow\theta)$
Distribution	$\theta\wedge(\phi_1\vee ...\vee\phi_n)$	$(\theta\wedge\phi_1)\vee ...\vee(\theta\wedge\phi_n)$
	$(\theta\wedge\phi_1)\vee ...\vee(\theta\wedge\phi_n)$	$\theta\wedge(\phi_1\vee ...\vee\phi_n)$
	$(\phi_1\vee ...\vee\phi_n)\wedge\theta$	$(\phi_1\wedge\theta)\vee ...\vee(\phi_n\wedge\theta)$
	$(\phi_1\wedge\theta)\vee ...\vee(\phi_n\wedge\theta)$	$(\phi_1\vee ...\vee\phi_n)\wedge\theta$
	$\theta\vee(\phi_1\wedge ...\wedge\phi_n)$	$(\theta\vee\phi_1)\wedge ...\wedge(\theta\vee\phi_n)$
	$(\theta\vee\phi_1)\wedge ...\wedge(\theta\vee\phi_n)$	$\theta\vee(\phi_1\wedge ...\wedge\phi_n)$
	$(\phi_1\wedge ...\wedge\phi_n)\vee\theta$	$(\phi_1\vee\theta)\wedge ...\wedge(\phi_n\vee\theta)$
	$(\phi_1\vee\theta)\wedge ...\wedge(\phi_n\vee\theta)$	$(\phi_1\wedge ...\wedge\phi_n)\vee\theta$
Set 4: Basic Rules of GQD, table 7.1 (on page 163)		
$\forall$-*Elim*	$\forall\beta\phi$	ϕa/β, for 'a' any individual constant
$\forall$-*Intro*	ϕa/β	$\forall\beta\phi$, iff 'a' does not occur in ϕ nor in any unboxed assumption
$\exists$-*Intro*	ϕa/β	$\exists\beta\phi$
$\exists$-*Elim*	$\exists\beta\phi$, ϕa/$\beta\rightarrow\theta$	θ, iff 'a' does not occur in ϕ or θ, nor in any unboxed assumption
Set 5: Exchange Shortcut Rules of GQD, table 7.19 (on page 170)		
QN	$\sim\forall\beta\phi$	$\exists\beta\sim\phi$
	$\exists\beta\sim\phi$	$\sim\forall\beta\phi$
	$\sim\exists\beta\phi$	$\forall\beta\sim\phi$
	$\forall\beta\sim\phi$	$\sim\exists\beta\phi$
Set 6: DNF Exchange Shortcut Rules for GSD, table 8.1 (on page 187)		
$\leftrightarrow$-*Exchange*	$\theta\leftrightarrow\psi$	$(\theta\wedge\psi)\vee(\sim\theta\wedge\sim\psi)$
	$(\theta\wedge\psi)\vee(\sim\theta\wedge\sim\psi)$	$\theta\leftrightarrow\psi$

Name	Given	May Add
Set 7: Prenex Exchange Shortcut Rules for GQD, table 8.5 (on page 193)		
α/β-Exch	$(\#\alpha)\phi$	$(\#\beta)\phi\beta/\alpha$
Q Shuffling	$((\#x)\theta \wedge \psi)$	$(\#x)(\theta \wedge \psi)$
	$(\theta \wedge (\#x)\psi)$	$(\#x)(\theta \wedge \psi)$
	$((\#x)\theta \vee \psi)$	$(\#x)(\theta \vee \psi)$
	$(\theta \vee (\#x)\psi)$	$(\#x)(\theta \vee \psi)$
	$(\theta \rightarrow (\#x)\psi)$	$(\#x)(\theta \rightarrow \psi)$
	$(\exists x\theta \rightarrow \psi)$	$\forall x(\theta \rightarrow \psi)$
	$(\forall x\theta \rightarrow \psi)$	$\exists x(\theta \rightarrow \psi)$
Set 8: The Method Shortcut Rules for GQD, table 8.6 (on page 202)		
Greg's Rule	$\psi_1 \vee ... \vee \psi_n$, where some $\psi_i = \phi_1 \wedge ... \wedge \phi_j \wedge ... \wedge \sim\phi_j \wedge ... \wedge \phi_m$	$\psi_1 \vee ... \vee \psi_{i-1} \vee \psi_{i+1} \vee ... \vee \psi_n$
$\vee/\wedge$-Elim	$\psi_1 \vee ... \vee \psi_n$, where each ψ_i contains ϕ	ϕ
OBA	$\phi_1 \wedge ... \wedge \phi_i \wedge ... \wedge \phi_n$, $\sim\phi_i$	$\sim(\phi_1 \wedge ... \wedge \phi_i \wedge ... \wedge \phi_n)$
	$\phi_1 \wedge ... \wedge \sim\phi_i \wedge ... \wedge \phi_n$, ϕ_i	$\sim(\phi_1 \wedge ... \wedge \sim\phi_i \wedge ... \wedge \phi_n)$
Set 9: Additonal Exchange Rules for QD, table 8.7 (on page 208)		
$\forall$-Shuffle	$\forall\alpha(\phi_1 \wedge ... \wedge \phi_n)$	$\forall\alpha\phi_1 \wedge ... \wedge \forall\alpha\phi_n$
$\exists$-Shuffle	$\exists\alpha(\phi_1 \vee ... \vee \phi_n)$	$\exists\alpha\phi_1 \vee ... \vee \exists\alpha\phi_n$
Q-Deletion	$\#\alpha\phi$	ϕ, iff α does not occur in ϕ
$\wedge$-Shuffle	$\phi_1 \wedge ... \wedge \phi_k \wedge ... \wedge \phi_n$	$\phi_k \wedge \phi_1 \wedge ... \wedge \phi_n$
$\vee$-Shuffle	$\phi_1 \vee ... \vee \phi_k \vee ... \vee \phi_n$	$\phi_k \vee \phi_1 \vee ... \vee \phi_n$
Set 10: Basic Rules for S5, table 9.1 (on page 223)		
$\Box$-Elim	$\Box\phi$	ϕ
$\Box$-Intro	ϕ (*)	$\Box\phi$
$\Diamond$-Elim	$\Diamond\phi$, $\phi \rightarrow \psi$ (*), (**)	ψ

Name	Given	May Add
$\Diamond$-*Intro*	ϕ	$\Diamond\,\phi$
Set 11: Modal Negation Exchange Shortcut Rules for S5, table 9.2 (on page 224)		
MN	$\sim\Box\,\phi$	$\Diamond\sim\phi$
	$\sim\Diamond\,\phi$	$\Box\sim\phi$
	$\sim\Diamond\sim\phi$	$\Box\,\phi$
	$\sim\Box\sim\phi$	$\Diamond\,\phi$
Set 12: Basic Rules for GQDI, 9.3 (on page 230)		
$=$-*Intro*		$t = t$
$=$-*Elim*	ϕ, $t = s$	$\phi t/s$

(*) in $\Box$-*Intro* and $\Diamond$-*Elim*, the rule can only be applied if all the open assumptions have modal prefixes.
(**) In $\Diamond$-*Elim*, the rule can only be applied if ψ has a modal prefix.

Works Cited

Arló-Costa, Horacio and Pacuit, Eric. 2006. "First-Order Classical Modal Logic". *Studia Logica* 84 (Special Issue Ways of Worlds II. On Possible Worlds and Related Notions):171–210.

Ballarin, Roberta. 2010. "Modern Origins of Modal Logic". In Edward N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. Winter 2010 edition. `http://plato.stanford.edu/archives/win2010/entries/logic-modal-origins/`.

Barwise, Jon and Etchemendy, John. 1987. *The Liar: an Essay on Truth and Circularity*. New York: Oxford University Press.

Bealer, George. 1998. "Propositions". *Mind* 107:1–32.

Beall, J.C. and van Fraassen, Bas. 2003. *Possibilities and Paradox: an Introduction to Modal and Many-Valued Logic*. Oxford New York: Oxford University Press.

Bergmann, Merrie, Moor, James, and Nelson, Jack. 2003. *The logic book*. New York London: McGraw-Hill Higher Education McGraw-Hill, 4 edition.

Blanchette, Patricia. 2001. "Logical Consequence". In Lou Goble (ed.), *The Blackwell Guide to Philosophical Logic*. Blackwell, 115–135.

Boole, George. 1854. *An Investigation of The Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*. Originally published by Macmillan. Reprint by Dover, 1958.

Borkowski, L. 1970. *Jan Lukasiewicz: Selected Works*. North-Holland Publishing Company.

Bull, R. A. and Segerberg, K. 2001. "Basic Modal Logic". In Dov M. Gabbay and Franz Guenthner (eds.), *Handbook of Philosophical Logic*, volume 3. Kluwer Academic Publishers, 1–82.

Carnap, Rudolf. 1946. "Modalities and Quantification". *Journal of Symbolic Logic* 11:33–64.

——. 1947. *Meaning and Necessity: A Study in Semantics and Modal Logic*. The University of Chicago Press.

Chomsky, Noam. 2002 [1957]. *Syntactic Structures*. Mouton de Gruyer (formerly Mouton, The Hague).

Church, Alonzo. 1956. *Introduction to Mathematical Logic*, volume 1. Princeton University Press.

Cocchiarella, Nino and Freund, Max. 2008. *Modal Logic: an Introduction to its Syntax and Semantics*. Oxford New York: Oxford University Press.

Cresswell, M. J. 2001. "Modal Logic". In Lou Goble (ed.), *The Blackwell Guide to Philosophical Logic*. Blackwell, 136–158.

De Morgan, Augustus. 1847. *Formal Logic: or The Calculus of Inference*. Taylor & Walton.

——. 1860. *Syllabus of a Proposed System of Logic*. Walton & Malbery.

Demopoulos, William and Clark, Peter. 2005. "The Logicism of Frege, Dedekind, and Russell". In Stewart Shapiro (ed.), *The Oxford Handbook of Philosophy of Mathematics and Logic*. Oxford University Press, 129–165.

Dipert, Randall. 1984. "Studies in Logic by Members of the Johns Hopkins University by Charles S. Peirce: Max H. Fisch; Achim Eschbach (Review)". *Transactions of the Charles S. Peirce Society* 20:469–472.

Dunn, J M and Belnap, N D. 1968. "The Substitution Interpretation of the Quantifiers". *Noûs* 2:177–185.

Ebbinghaus, H. D. 1985. "Extended Logics: The General Framework". In J. Barwise and S. Feferman (eds.), *Model-Theoretic Logics*, Perspectives ini Mathematical Logic. Springer-Verlag, 25–76.

Enderton, Herbert B. 2010. *Computability Theory: An Introduction to Recursion Theory*. Academic Press (Elsevier Science).

Fitch, Frederic. 1952. *Symbolic Logic*. Ronald Press.

Frege, Gottlob. 1879. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle. Trans. "Concept Script, a formal language of pure thought modelled upon that of arithmetic".

——. 1892. "Über Sinn und Bedeutung". *Zeitschrift für Philosophie und philosophische Kritik* 100:25–50. Trans. "On Sense and Reference", available in Martinich, A. P. (ed.), 2001, *The Philosophy of Language*, Oxford: Oxford University Press, 4th edition and in Peter Geach and Max Black (eds.), 1966, *Translations from the Philosophical Writings of Gottlob Frege*, Basil Blackwel, 2nd edition.

——. 1893/1903. *Grundgesetze der Arithmetik*. Jena: Verlag Hermann Pohle. Trans. "The Basic Laws of Arithmetic".

——. 1966 [1891]. "Function and Concept (Funktion und Begriff)". In *Translations from the Philosophical Writings of Gottlob Frege*. Basil Blackwell. Orig. title "Funktion und Begriff".

Garson, J. 2001. "Quantification in Modal Logic". In Dov M. Gabbay and Franz Guenthner (eds.), *Handbook of Philosophical Logic*, volume 3. Kluwer Academic Publishers, 267–324.

Gentzen, Gerhard. 1934. "Untersuchungen über das Logishe Schliessen". *Math Zeitschrift* 39:176–210 and 405–431.

Gödel, Kurt. 1929. *Über die Vollständigkeit des Logikkalküls*. Ph.D. thesis, University Of Vienna.

——. 1930. "Die Vollständigkeit der Axiome des logischen Funktionenkalküls". *Monatshefte für Mathematik und Physik* 37:349–360.

Goldblatt, Rob. 2006. "Mathematical Modal Logic: A View of its Evolution". In Dov M. Gabbay and John Woods (eds.), *Handbook of the History of Logic*, volume 7. Elsevier, 1–98.

Grandy, Richard. 1993. "What do 'Q' and 'R' Stand for Anyway?" In R.I.G. Hughes (ed.), *A Philosophical Companion to First-Order Logic*. Hackett, 50–61.

Gupta, Anil. 2001. "Truth". In Lou Goble (ed.), *The Blackwell Guide to Philosophical Logic*. Blackwell, 90–114.

Henkin, Leon. 1949. "The Completeness of the First-Order Functional Calculus". *Journal of Symbolic Logic* 14:159–166.

Hilbert, David and Ackermann, Wilhelm. 1928. *Grundzüge der theoretischen Logik*. Springer-Verlag.

Hintikka, Jaakko. 1961. "Modalities and Quantification". *Theoria* 27:119–28.

——. 1996. *The principles of mathematics revisited*. Cambridge New York: Cambridge University Press.

Hodges, Wilfrid. 1997. *A shorter model theory*. Cambridge New York: Cambridge University Press.

——. 2001a. "Classical Logic I: First-Order Logic". In Lou Goble (ed.), *The Blackwell Guide to Philosophical Logic*. Blackwell, 9–32.

——. 2001b. "Elementary Predicate Logic". In Dov M. Gabbay and Franz Guenthner (eds.), *Handbook of Philosophical Logic*, volume 1. Kluwer Academic Publishers, 1–129.

Hughes, G. and Cresswell, M.J. 1996. *A New Introduction to Modal Logic*. London New York: Routledge.

Jacquette, Dale (ed.). 2002. *Philosophy of Logic: an anthology*. Blackwell Philosophy Anthologies. Malden, Mass: Blackwell Publishers.

Jaśkowski, Stanisław. 1934. "On the Rules of Suppositions in Formal Logic". *Studia Logica* 1:5–32.

Kalish, Donald and Montague, Richard. 1964. *Logic: Techniques of Formal Reasoning*. Harcourt, Brace and World.

Kanger, Stig. 1957. "Provability in Logic". In *Acta Universitatis Stockholmiensis*, volume 1 of *Stockholm Studies in Philosophy*. Almqvist and Wiksell.

King, Jeffrey C. 2007. *The Nature and Structure of Content*. Oxford University Press.

Kleene, Stephen. 2002 [1967]. *Mathematical logic*. Mineola, N.Y: Dover Publications.

Kripke, Saul. 1959. "A Completeness Theorem in Modal Logic". *Journal of Symbolic Logic* 24:1–14.

——. 1963a. "Semantical Analysis of Modal Logic I: Normal Modal Propositional Calculi". *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 9:67–96.

——. 1963b. "Semantical Considerations on Modal Logic". *Acta Philosophica Fennica* 16:83–94.

——. 1965. "Semantical Analysis of Modal Logic II. Non-normal Modal Propositional Calculi". In J. W. Addison, L. Henkin, and A. Tarski (eds.), *Symposium on the Theory of Models*. Amsterdam: North-Holland, 206–220.

——. 1975. "An Outline of a Theory of Truth". *Journal of Philosophy* 72:690–716.

Ladd-Franklin, Christine. 1883. "On the Algebra of Logic". In C. S. Peirce (ed.), *Studies in Logic*. Little, Brown and Co, 17–71.

Leblanc, Hugues. 2001. "Alternative to Standard First-order Semantics". In Dov M. Gabbay and Franz Guenthner (eds.), *Handbook of Philosophical Logic*, volume 2. Kluwer Academic Publishers, 53–132.

Lewis, C. I. and Langford, Cooper H. 1932. *Symbolic Logic*. London: Century. 2nd edition 1959, New York: Dover.

Löwenheim, Leopold. 1915. "Über Möglichkeiten im Relativkallkül". *Math Annalen* 76:447–470.

Marcus, Ruth Barcan. 1946a. "The Deduction Theorem in a Functional Calculus of First Order Based on Strict Implication". *Journal of Symbolic Logic* 11:115–118.

——. 1946b. "A Functional Calculus of First Order Based on Strict Implication". *Journal of Symbolic Logic* 11:1–16.

——. 1947. "The Identity of Individuals in a Strict Functional Calculus of Second Order". *Journal of Symbolic Logic* 12:12–15.

——. 1993. *Modalities: Philosophical Essays*. Oxford University Press.

Mates, Benson. 1972. *Elementary Logic*. Oxford University Press, 2 edition.

Mitchell, O H. 1883. "On a New Algebra of Logic". In C. S. Peirce (ed.), *Studies in Logic*. Little, Brown and Co, 72–106.

Montague, Richard. 1960. "Logical Necessity, Physical Necessity, Ethics, and Quantifiers". *Inquiry* 3:259–269.

——. 1970. "Universal Grammar". *Theoria* 36:373–98.

Peirce, C.S. 1883. "A Theory of Probable Inference. Note B. The Logic of Relatives". In Boston Members of the Johns Hopkins University (ed.), *Studies in Logic*. Little, Brown and Co.

——. 1902. "The Simplest Mathematics". In C Hartshorne et al (ed.), *Collected Papers of Charles Sanders Peirce*, volume IV. Harvard University Press, 189–262.

Pospesel, Howard and Marans, David. 1978. *Arguments: Deductive Logic Exercises*. Pearson Education, Inc, 2 edition.

Post, Emil. 1921. "Introduction to a General Theory of Elementary Propositions". *American Journal of Mathematics* 43:163–185.

Prior, A.N. 1957. *Time and Modality*. Clarendon Press.

Quine, Willard Van Orman. 1950. *Methods of Logic*. Holt.

——. 1982. *Methods of Logic*. Harvard University Press, 4 edition.

——. 1986. *Philosophy of logic*. Cambridge, Mass: Harvard University Press, 2 edition.

Russell, Bertrand. 1996 [1903]. *The Principles of Mathematics*. New York: W.W. Norton & Company.

Schiffer, Stephen. 1987. *Remnants of Meaning*. The MIT Press.

Scott, Dana. 1970. "Advice in Modal Logic". In K. Lambert (ed.), *Philosophical Problems in Logic*. Dordrecht, Netherlands: Reidel, 143–73.

Shapiro, Stewart. 2005. "Logical Consequence, Proof Theory, and Modal Theory". In Stewart Shapiro (ed.), *The Oxford Handbook of Philosophy of Mathematics and Logic*. Oxford University Press, 651–670.

Skolem, Thoralf. 1920. "Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit Mathematischer Sätze nebst einem Theoreme über dichte Mengen". *Videnskapsselskapets Skrifter, I. Matem.-Naturv. Klasse* 4.

——. 1922. "Einige Bemerkungen zur axiomatischen Bergründung der Mengenlehre". *Matematikerkongressen i Helsingfors den 4–7 Juli 1922* .

——. 1967 [1928]. "On mathematical logic". In Jean van Heijenoort (ed.), *From Frege to Gödel: A Source Book in Mathematical Logic, 1879 to 1931*. Harvard University Press, 508–524.

Smith, Nicholas J.J. 2012. *Logic: The Laws of Truth*. Princeton University Press.

Smith, Robin. 1995. "Logic". In Jonathan Barnes (ed.), *The Cambridge Companion to Aristotle*. Cambridge University Press, 27–65.

Smullyan, Raymond M. and Fitting, Melvin. 2010. *Set Theory and the Continuum Problem*. Dover. Originally published in 1996, Oxford University Press.

Soames, Scott. 2010. *What is Meaning?* Soochow University Lectures in Philosophy. Princeton University Press.

Stevenson, L. 1973. "Frege's Two Definitions of Quantification". *Philosophical Quarterly* 23:207–223.

Tarski, Alfred. 1944. "The Semantic Conception of Truth and the Foundations of Semantics". *Philosophy and Phenomenological Research* 4:341–375. Available in Martinich, A. P. (ed.), 2001, *The Philosophy of Language*, Oxford: Oxford University Press, 4th edition.

——. 1983 [1933]. "The Concept of Truth in Formalized Languages". In John Corcoran (ed.), *Logic, Semantics, Metamathematics: Papers from 1923 to 1938*. Hackett Publishing, 152–278.

Tarski, Alfred and Vaught, R. L. 1956. "Arithmetical Extensions of Relational Systems". *Compositio Math* 13:81–102.

Vaught, R. L. 1974. "Model Theory before 1945". In L. Henkin et al (ed.), *Proceedings of the Tarski Symposium*. AMS, 153–172.

Westerståhl, Dag. 2001. "Quantifiers". In Lou Goble (ed.), *The Blackwell Guide to Philosophical Logic*. Blackwell, 437–460.

Zakharyaschev, M., Wolter, F., and Chagrov, A. 2001. "Advanced Modal Logic". In Dov M. Gabbay and Franz Guenthner (eds.), *Handbook of Philosophical Logic*, volume 3. Kluwer Academic Publishers, 83–266.