forallχ

An Introduction to Formal Logic

P.D. Magnus
University at Albany, State University of New York

Edited by Nihal Anand
University of Texas at Dallas

fecundity.com/logic, version 1.4 [240226]
This book is offered under a Creative Commons license.
(Attribution 4.0)

The author would like to thank the people who made this project possible. Notable among these are Cristyn Magnus, who read many early drafts; Aaron Schiller, who was an early adopter and provided considerable, helpful feedback; and Bin Kang, Craig Erb, Nathan Carter, Wes McMichael, Selva Samuel, Dave Krueger, Brandon Lee, Toan Tran, Marcus Adams, Matthew Brown, and the students of Introduction to Logic, who detected various errors in previous versions of the book.

© 2005–2024 by P.D. Magnus. Some rights reserved.

You are free to copy this book, to distribute it, to display it, and to make derivative works, under the following condition: Attribution. You must give the original author credit. — For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder. Your fair use and other rights are in no way affected by the above. — This is a human-readable summary of the full license, which is available on-line at `http://creativecommons.org/licenses/by/4.0/`

Typesetting was carried out entirely in LaTeX2ε. The style for typesetting proofs is based on fitch.sty (v0.4) by Peter Selinger, University of Ottawa.

This copy of forall$\mathcal{x}$ is current as of February 26, 2024. The most recent version is available on-line at `https://github.com/NihalBAnand/forallx_UTD/releases/latest/download/forallx.pdf`

forallχ contents

1 What is logic? **5**
1.1 Arguments . . . 5
1.2 Sentences . . . 6
1.3 Two ways that arguments can go wrong . . . 7
1.4 Deductive validity . . . 8
1.5 Other logical notions . . . 10
1.6 Formal languages . . . 12
Practice Exercises . . . 14

2 Sentential logic **16**
2.1 Sentence letters . . . 16
2.2 Connectives . . . 18
2.3 Other symbolization . . . 26
2.4 Sentences of SL . . . 27
Practice Exercises . . . 31

3 Truth tables **35**
3.1 Truth-functional connectives . . . 35
3.2 Complete truth tables . . . 36
3.3 Using truth tables . . . 39
3.4 Partial truth tables . . . 40
Practice Exercises . . . 42

4 Proofs in SL **46**
4.1 Basic rules for SL . . . 47
4.2 Derived rules . . . 56
4.3 Rules of replacement . . . 57
4.4 Proof-theoretic concepts . . . 59
Practice Exercises . . . 60

5 Quantified logic **63**
5.1 From sentences to predicates . . . 63
5.2 Building blocks of QL . . . 65
5.3 Quantifiers . . . 69
5.4 Translating to QL . . . 72

5.5 Sentences of QL 82
5.6 Identity 85
Practice Exercises 89

6 Formal semantics **96**
6.1 Semantics for SL 97
6.2 Interpretations and models in QL 101
6.3 Semantics for identity 105
6.4 Working with models 106
6.5 Truth in QL 111
Practice Exercises 111

7 Proofs in QL **115**
7.1 Rules for quantifiers 115
7.2 Rules for identity 121
7.3 Proof strategy 122
7.4 Proofs and models 124
7.5 Soundness and completeness 125
Practice Exercises 127

A Other symbolic notation **131**

B Solutions to selected exercises **134**

C Quick Reference **147**

Chapter 1

What is logic?

Logic is the business of evaluating arguments, sorting good ones from bad ones. In everyday language, we sometimes use the word 'argument' to refer to belligerent shouting matches. If you and a friend have an argument in this sense, things are not going well between the two of you.

In logic, we are not interested in the teeth-gnashing, hair-pulling kind of argument. A logical argument is structured to give someone a reason to believe some conclusion. Here is one such argument:

(1) It is raining heavily.
(2) If you do not take an umbrella, you will get soaked.
$\therefore$ You should take an umbrella.

The three dots on the third line of the argument mean 'Therefore' and they indicate that the final sentence is the *conclusion* of the argument. The other sentences are *premises* of the argument. If you believe the premises, then the argument provides you with a reason to believe the conclusion.

This chapter discusses some basic logical notions that apply to arguments in a natural language like English. It is important to begin with a clear understanding of what arguments are and of what it means for an argument to be valid. Later we will translate arguments from English into a formal language. We want formal validity, as defined in the formal language, to have at least some of the important features of natural-language validity.

1.1 Arguments

When people mean to give arguments, they typically often use words like 'therefore' and 'because.' When analyzing an argument, the first thing to do is to separate the premises from the conclusion. Words like these are a clue to what the argument is supposed to be, especially if— in the argument as given— the conclusion comes at the beginning or in the middle of the argument.

premise indicators: since, because, given that

conclusion indicators: therefore, hence, thus, then, so

To be perfectly general, we can define an ARGUMENT as a series of sentences. The sentences at the beginning of the series are premises. The final sentence in the series is the conclusion. If the premises are true and the argument is a good one, then you have a reason to accept the conclusion.

Notice that this definition is quite general. Consider this example:

There is coffee in the coffee pot.
There is a dragon playing bassoon on the armoire.
$\therefore$ Salvador Dali was a poker player.

It may seem odd to call this an argument, but that is because it would be a terrible argument. The two premises have nothing at all to do with the conclusion. Nevertheless, given our definition, it still counts as an argument— albeit a bad one.

1.2 Sentences

In logic, we are only interested in sentences that can figure as a premise or conclusion of an argument. So we will say that a SENTENCE is something that can be true or false.

You should not confuse the idea of a sentence that can be true or false with the difference between fact and opinion. Often, sentences in logic will express things that would count as facts— such as 'Kierkegaard was a hunchback' or 'Kierkegaard liked almonds.' They can also express things that you might think of as matters of opinion— such as, 'Almonds are yummy.'

Also, there are things that would count as 'sentences' in a linguistics or grammar course that we will not count as sentences in logic.

Questions In a grammar class, 'Are you sleepy yet?' would count as an interrogative sentence. Although you might be sleepy or you might be alert, the question itself is neither true nor false. For this reason, questions will not count as sentences in logic. Suppose you answer the question: 'I am not sleepy.'

This is either true or false, and so it is a sentence in the logical sense. Generally, *questions* will not count as sentences, but *answers* will.

'What is this course about?' is not a sentence. 'No one knows what this course is about' is a sentence.

Imperatives Commands are often phrased as imperatives like 'Wake up!', 'Sit up straight', and so on. In a grammar class, these would count as imperative sentences. Although it might be good for you to sit up straight or it might not, the command is neither true nor false. Note, however, that commands are not always phrased as imperatives. 'You will respect my authority' *is* either true or false— either you will or you will not— and so it counts as a sentence in the logical sense.

Exclamations 'Ouch!' is sometimes called an exclamatory sentence, but it is neither true nor false. We will treat 'Ouch, I hurt my toe!' as meaning the same thing as 'I hurt my toe.' The 'ouch' does not add anything that could be true or false.

1.3 Two ways that arguments can go wrong

Consider the argument that you should take an umbrella (on p. 5, above). If premise (1) is false— if it is sunny outside— then the argument gives you no reason to carry an umbrella. Even if it is raining outside, you might not need an umbrella. You might wear a rain poncho or keep to covered walkways. In these cases, premise (2) would be false, since you could go out without an umbrella and still avoid getting soaked.

Suppose for a moment that both the premises are true. You do not own a rain poncho. You need to go places where there are no covered walkways. Now does the argument show you that you should take an umbrella? Not necessarily. Perhaps you enjoy walking in the rain, and you would like to get soaked. In that case, even though the premises were true, the conclusion would be false.

For any argument, there are two ways that it could be weak. First, one or more of the premises might be false. An argument gives you a reason to believe its conclusion only if you believe its premises. Second, the premises might fail to support the conclusion. Even if the premises were true, the form of the argument might be weak. The example we just considered is weak in both ways.

When an argument is weak in the second way, there is something wrong with the *logical form* of the argument: Premises of the kind given do not necessarily lead to a conclusion of the kind given. We will be interested primarily in the logical form of arguments.

Consider another example:

You are reading this book.
This is a logic book.
∴ You are a logic student.

This is not a terrible argument. Most people who read this book are logic students. Yet, it is possible for someone besides a logic student to read this book. If your roommate picked up the book and thumbed through it, they would not immediately become a logic student. So the premises of this argument, even though they are true, do not guarantee the truth of the conclusion. Its logical form is less than perfect.

An argument that had no weakness of the second kind would have perfect logical form. If its premises were true, then its conclusion would *necessarily* be true. We call such an argument 'deductively valid' or just 'valid.'

Even though we might count the argument above as a good argument in some sense, it is not valid; that is, it is 'invalid.' One important task of logic is to sort valid arguments from invalid arguments.

1.4 Deductive validity

An argument is deductively VALID if and only if it is impossible for the premises to be true and the conclusion false.

The crucial thing about a valid argument is that it is impossible for the premises to be true *at the same time* that the conclusion is false. Consider this example:

Oranges are either fruits or musical instruments.
Oranges are not fruits.
∴ Oranges are musical instruments.

The conclusion of this argument is ridiculous. Nevertheless, it follows validly from the premises. This is a valid argument. *If* both premises were true, *then* the conclusion would necessarily be true.

This shows that a deductively valid argument does not need to have true premises or a true conclusion. Conversely, having true premises and a true conclusion is not enough to make an argument valid. Consider this example:

London is in England.

Beijing is in China.

∴ Paris is in France.

The premises and conclusion of this argument are, as a matter of fact, all true. This is a terrible argument, however, because the premises have nothing to do with the conclusion. Imagine what would happen if Paris declared independence from the rest of France. Then the conclusion would be false, even though the premises would both still be true. Thus, it is *logically possible* for the premises of this argument to be true and the conclusion false. The argument is invalid.

The important thing to remember is that validity is not about the actual truth or falsity of the sentences in the argument. Instead, it is about the form of the argument: The truth of the premises is incompatible with the falsity of the conclusion.

Inductive arguments

There can be good arguments which nevertheless fail to be deductively valid. Consider this one:

In January 1997, it rained in San Diego.

In January 1998, it rained in San Diego.

In January 1999, it rained in San Diego.

∴ It rains every January in San Diego.

This is an INDUCTIVE argument, because it generalizes from many cases to a conclusion about all cases.

Certainly, the argument could be made stronger by adding additional premises: In January 2000, it rained in San Diego. In January 2001... and so on. Regardless of how many premises we add, however, the argument will still not be deductively valid. It is possible, although unlikely, that it will fail to rain next January in San Diego. Moreover, we know that the weather can be fickle. No amount of evidence should convince us that it rains there *every* January. Who is to say that some year will not be a freakish year in which there is no rain in January in San Diego; even a single counter-example is enough to make the conclusion of the argument false.

Inductive arguments, even good inductive arguments, are not deductively valid. We will not be interested in inductive arguments in this book.

1.5 Other logical notions

In addition to deductive validity, we will be interested in some other logical concepts.

Truth-values

True or false is said to be the TRUTH-VALUE of a sentence. We defined sentences as things that could be true or false; we could have said instead that sentences are things that can have truth-values.

Logical truth

In considering arguments formally, we care about what would be true *if* the premises were true. Generally, we are not concerned with the actual truth value of any particular sentences— whether they are *actually* true or false. Yet there are some sentences that must be true, just as a matter of logic.

Consider these sentences:

1. It is raining.
2. Either it is raining, or it is not.
3. It is both raining and not raining.

In order to know if sentence 1 is true, you would need to look outside or check the weather channel. Logically speaking, it might be either true or false. Sentences like this are called *contingent* sentences.

Sentence 2 is different. You do not need to look outside to know that it is true. Regardless of what the weather is like, it is either raining or not. This sentence is *logically true*; it is true merely as a matter of logic, regardless of what the world is actually like. A logically true sentence is called a TAUTOLOGY.

You do not need to check the weather to know about sentence 3, either. It must be false, simply as a matter of logic. It might be raining here and not raining across town, it might be raining now but stop raining even as you read this, but it is impossible for it to be both raining and not raining here at this moment. The third sentence is *logically false*; it is false regardless of what the world is like. A logically false sentence is called a CONTRADICTION.

To be precise, we can define a CONTINGENT SENTENCE as a sentence that is neither a tautology nor a contradiction.

A sentence might *always* be true and still be contingent. For instance, if there never were a time when the universe contained fewer than seven things, then the sentence 'At least seven things exist' would always be true. Yet the sentence is contingent; its truth is not a matter of logic. There is no contradiction in considering a possible world in which there are fewer than seven things. The important question is whether the sentence *must* be true, just on account of logic.

Logical equivalence

We can also ask about the logical relations *between* two sentences. For example:

> John went to the store after he washed the dishes.
> John washed the dishes before he went to the store.

These two sentences are both contingent, since John might not have gone to the store or washed dishes at all. Yet they must have the same truth-value. If either of the sentences is true, then they both are; if either of the sentences is false, then they both are. When two sentences necessarily have the same truth value, we say that they are LOGICALLY EQUIVALENT.

Consistency

Consider these two sentences:

> **B1** My only brother is taller than I am.
> **B2** My only brother is shorter than I am.

Logic alone cannot tell us which, if either, of these sentences is true. Yet we can say that *if* the first sentence (B1) is true, *then* the second sentence (B2) must be false. And if B2 is true, then B1 must be false. It cannot be the case that both of these sentences are true.

If a set of sentences could not all be true at the same time, like B1–B2, they are said to be INCONSISTENT. Otherwise, they are CONSISTENT.

We can ask about the consistency of any number of sentences. For example, consider the following list of sentences:

> **G1** There are at least four giraffes at the wild animal park.
> **G2** There are exactly seven gorillas at the wild animal park.
> **G3** There are not more than two martians at the wild animal park.
> **G4** Every giraffe at the wild animal park is a martian.

G1 and G4 together imply that there are at least four martian giraffes at the park. This conflicts with G3, which implies that there are no more than two martian giraffes there. So the set of sentences G1–G4 is inconsistent. Notice that the inconsistency has nothing at all to do with G2. G2 just happens to be part of an inconsistent set.

Sometimes, people will say that an inconsistent set of sentences 'contains a contradiction.' By this, they mean that it would be logically impossible for all of the sentences to be true at once. A set might be inconsistent even if each of the sentences in it is either contingent or tautologous. When a single sentence is a contradiction, then that sentence alone cannot be true.

1.6 Formal languages

Here is a famous valid argument:

Socrates is a man.
All men are mortal.
∴ Socrates is mortal.

This is an iron-clad argument. The only way you could challenge the conclusion is by denying one of the premises— the logical form is impeccable. What about this next argument?

Socrates is a man.
All men are carrots.
∴ Socrates is a carrot.

This argument might be less interesting than the first, because the second premise is obviously false. There is no clear sense in which all men are carrots. Yet the argument is valid. To see this, notice that both arguments have this form:

S is M.
All Ms are Cs.
∴ S is C.

In both arguments S stands for Socrates and M stands for man. In the first argument, C stands for mortal; in the second, C stands for carrot. Both arguments have this form, and every argument of this form is valid. So both arguments are valid.

What we did here was replace words like 'man' or 'carrot' with symbols like 'M' or 'C' so as to make the logical form explicit. This is the central idea behind formal logic. We want to remove irrelevant or distracting features of the argument to make the logical form more perspicuous.

Starting with an argument in a *natural language* like English, we translate the argument into a *formal language*. Parts of the English sentences are replaced with letters and symbols. The goal is to reveal the formal structure of the argument, as we did with these two.

There are formal languages that work like the symbolization we gave for these two arguments. A logic like this was developed by Aristotle, a philosopher who lived in Greece during the 4th century BC. Aristotle was a student of Plato and the tutor of Alexander the Great. Aristotle's logic, with some revisions, was the dominant logic in the western world for more than two millennia.

In Aristotelean logic, categories are replaced with capital letters. Every sentence of an argument is then represented as having one of four forms, which medieval logicians labeled in this way: (A) All As are Bs. (E) No As are Bs. (I) Some A is B. (O) Some A is not B.

It is then possible to describe valid *syllogisms*, three-line arguments like the two we considered above. Medieval logicians gave mnemonic names to all of the valid argument forms. The form of our two arguments, for instance, was called *Barbara*. The vowels in the name, all As, represent the fact that the two premises and the conclusion are all (A) form sentences.

There are many limitations to Aristotelean logic. One is that it makes no distinction between kinds and individuals. So the first premise might just as well be written 'All Ss are Ms': All Socrateses are men. Despite its historical importance, Aristotelean logic has been superceded. The remainder of this book will develop two formal languages.

The first is SL, which stands for *sentential logic*. In SL, the smallest units are sentences themselves. Simple sentences are represented as letters and connected with logical connectives like 'and' and 'not' to make more complex sentences.

The second is QL, which stands for *quantified logic*. In QL, the basic units are objects, properties of objects, and relations between objects.

When we translate an argument into a formal language, we hope to make its logical structure clearer. We want to include enough of the structure of the English language argument so that we can judge whether the argument is valid or invalid. If we included every feature of the English language, all of the subtlety and nuance, then there would be no advantage in translating to a formal language. We might as well think about the argument in English.

At the same time, we would like a formal language that allows us to represent many kinds of English language arguments. This is one reason to prefer QL to Aristotelean logic; QL can represent every valid argument of Aristotelean logic and more.

So when deciding on a formal language, there is inevitably a tension between wanting to capture as much structure as possible and wanting a simple formal language— simpler formal languages leave out more. This means that there is no perfect formal language. Some will do a better job than others in translating particular English-language arguments.

In this book, we make the assumption that *true* and *false* are the only possible truth-values. Logical languages that make this assumption are called *bivalent*, which means *two-valued*. Aristotelean logic, SL, and QL are all bivalent, but there are limits to the power of bivalent logic. For instance, some philosophers have claimed that the future is not yet determined. If they are right, then sentences about *what will be the case* are not yet true or false. Some formal languages accommodate this by allowing for sentences that are neither true nor false, but something in between. Other formal languages, so-called paraconsistent logics, allow for sentences that are both true *and* false.

The languages presented in this book are not the only possible formal languages. However, most nonstandard logics extend on the basic formal structure of the bivalent logics discussed in this book. So this is a good place to start.

Summary of logical notions

- ▷ An argument is (deductively) VALID if it is impossible for the premises to be true and the conclusion false; it is INVALID otherwise.
- ▷ A TAUTOLOGY is a sentence that must be true, as a matter of logic.
- ▷ A CONTRADICTION is a sentence that must be false, as a matter of logic.
- ▷ A CONTINGENT SENTENCE is neither a tautology nor a contradiction.
- ▷ Two sentences are LOGICALLY EQUIVALENT if they necessarily have the same truth value.
- ▷ A set of sentences is CONSISTENT if it is logically possible for all the members of the set to be true at the same time; it is INCONSISTENT otherwise.

Practice Exercises

At the end of each chapter, you will find a series of practice problems that review and explore the material covered in the chapter. There is no substitute for actually working through some problems, because logic is more about a way of thinking than it is about memorizing facts. The answers to some of the problems are provided at the end of the book in appendix B; the problems that are solved in the appendix are marked with a $\star$.

Part A Which of the following are 'sentences' in the logical sense?

1. England is smaller than China.
2. Greenland is south of Jerusalem.
3. Is New Jersey east of Wisconsin?
4. The atomic number of helium is 2.
5. The atomic number of helium is π.
6. I hate overcooked noodles.
7. Blech! Overcooked noodles!
8. Overcooked noodles are disgusting.
9. Take your time.
10. This is the last question.

Part B For each of the following: Is it a tautology, a contradiction, or a contingent sentence?

1. Caesar crossed the Rubicon.
2. Someone once crossed the Rubicon.
3. No one has ever crossed the Rubicon.
4. If Caesar crossed the Rubicon, then someone has.
5. Even though Caesar crossed the Rubicon, no one has ever crossed the Rubicon.
6. If anyone has ever crossed the Rubicon, it was Caesar.

⋆ **Part C** Look back at the sentences G1–G4 on p. 11, and consider each of the following sets of sentences. Which are consistent? Which are inconsistent?

1. G2, G3, and G4
2. G1, G3, and G4
3. G1, G2, and G4
4. G1, G2, and G3

⋆ **Part D** Which of the following is possible? If it is possible, give an example. If it is not possible, explain why.

1. A valid argument that has one false premise and one true premise
2. A valid argument that has a false conclusion
3. A valid argument, the conclusion of which is a contradiction
4. An invalid argument, the conclusion of which is a tautology
5. A tautology that is contingent
6. Two logically equivalent sentences, both of which are tautologies
7. Two logically equivalent sentences, one of which is a tautology and one of which is contingent
8. Two logically equivalent sentences that together are an inconsistent set
9. A consistent set of sentences that contains a contradiction
10. An inconsistent set of sentences that contains a tautology

Chapter 2

Sentential logic

This chapter introduces a logical language called SL. It is a version of *sentential logic*, because the basic units of the language will represent entire sentences.

2.1 Sentence letters

In SL, capital letters are used to represent basic sentences. Considered only as a symbol of SL, the letter A could mean any sentence. So when translating from English into SL, it is important to provide a *symbolization key*. The key provides an English language sentence for each sentence letter used in the symbolization.

For example, consider this argument:

There is an apple on the desk.
If there is an apple on the desk, then Jenny made it to class.
∴ Jenny made it to class.

This is obviously a valid argument in English. In symbolizing it, we want to preserve the structure of the argument that makes it valid. What happens if we replace each sentence with a letter? Our symbolization key would look like this:

A: There is an apple on the desk.
B: If there is an apple on the desk, then Jenny made it to class.
C: Jenny made it to class.

We would then symbolize the argument in this way:

A

B
$\therefore C$

There is no necessary connection between some sentence A, which could be any sentence, and some other sentences B and C, which could be any sentences. The structure of the argument has been completely lost in this translation.

The important thing about the argument is that the second premise is not merely *any* sentence, logically divorced from the other sentences in the argument. The second premise contains the first premise and the conclusion *as parts*. Our symbolization key for the argument only needs to include meanings for A and C, and we can build the second premise from those pieces. So we symbolize the argument this way:

A
If A, then C.
$\therefore C$

This preserves the structure of the argument that makes it valid, but it still makes use of the English expression 'If… then….' Although we ultimately want to replace all of the English expressions with logical notation, this is a good start.

The sentences that can be symbolized with sentence letters are called *atomic sentences*, because they are the basic building blocks out of which more complex sentences can be built. Whatever logical structure a sentence might have is lost when it is translated as an atomic sentence. From the point of view of SL, the sentence is just a letter. It can be used to build more complex sentences, but it cannot be taken apart.

There are only twenty-six letters of the alphabet, but there is no logical limit to the number of atomic sentences. We can use the same letter to symbolize different atomic sentences by adding a subscript, a small number written after the letter. We could have a symbolization key that looks like this:

$\mathbf{A_1}$**:** The apple is under the armoire.
$\mathbf{A_2}$**:** Arguments in SL always contain atomic sentences.
$\mathbf{A_3}$**:** Adam Ant is taking an airplane from Anchorage to Albany.
$\vdots$
$\mathbf{A_{294}}$**:** Alliteration angers otherwise affable astronauts.

Keep in mind that each of these is a different sentence letter. When there are subscripts in the symbolization key, it is important to keep track of them.

2.2 Connectives

Logical connectives are used to build complex sentences from atomic components. There are five logical connectives in SL. This table summarizes them, and they are explained below.

symbol	what it is called	what it means
$\neg$	negation	'It is not the case that...'
$\&$	conjunction	'Both... and ...'
$\vee$	disjunction	'Either... or ...'
$\rightarrow$	conditional	'If ... then ...'
$\leftrightarrow$	biconditional	'... if and only if ...'

Negation

Consider how we might symbolize these sentences:

1. Mary is in Barcelona.
2. Mary is not in Barcelona.
3. Mary is somewhere besides Barcelona.

In order to symbolize sentence 1, we will need one sentence letter. We can provide a symbolization key:

B: Mary is in Barcelona.

Note that here we are giving B a different interpretation than we did in the previous section. The symbolization key only specifies what B means *in a specific context*. It is vital that we continue to use this meaning of B so long as we are talking about Mary and Barcelona. Later, when we are symbolizing different sentences, we can write a new symbolization key and use B to mean something else.

Now, sentence 1 is simply B.

Since sentence 2 is obviously related to the sentence 1, we do not want to introduce a different sentence letter. To put it partly in English, the sentence means 'Not B.' In order to symbolize this, we need a symbol for logical negation. We will use '$\neg$.' Now we can translate 'Not B' to $\neg B$.

Sentence 3 is about whether or not Mary is in Barcelona, but it does not contain the word 'not.' Nevertheless, it is obviously logically equivalent to sentence 2. They both mean: It is not the case that Mary is in Barcelona. As such, we can translate both sentence 2 and sentence 3 as $\neg B$.

> A sentence can be symbolized as $\neg\mathcal{A}$ if it can be paraphrased in English as 'It is not the case that $\mathcal{A}$.'

Consider these further examples:

4. The widget can be replaced if it breaks.
5. The widget is irreplaceable.
6. The widget is not irreplaceable.

If we let R mean 'The widget is replaceable', then sentence 4 can be translated as R.

What about sentence 5? Saying the widget is irreplaceable means that it is not the case that the widget is replaceable. So even though sentence 5 is not negative in English, we symoblize it using negation as $\neg R$.

Sentence 6 can be paraphrased as 'It is not the case that the widget is irreplaceable.' Using negation twice, we translate this as $\neg\neg R$. The two negations in a row each work as negations, so the sentence means 'It is not the case that... it is not the case that... R.' If you think about the sentence in English, it is logically equivalent to sentence 4. So when we define logical equivalence in SL, we will make sure that R and $\neg\neg R$ are logically equivalent.

More examples:

7. Elliott is happy.
8. Elliott is unhappy.

If we let H mean 'Elliot is happy', then we can symbolize sentence 7 as H.

However, it would be a mistake to symbolize sentence 8 as $\neg H$. If Elliott is unhappy, then he is not happy— but sentence 8 does not mean the same thing as 'It is not the case that Elliott is happy.' It could be that he is not happy but that he is not unhappy either. Perhaps he is somewhere between the two. In order to allow for the possibility that he is indifferent, we would need a new sentence letter to symbolize sentence 8.

For any sentence $\mathcal{A}$: If $\mathcal{A}$ is true, then $\neg\mathcal{A}$ is false. If $\neg\mathcal{A}$ is true, then $\mathcal{A}$ is false. Using 'T' for true and 'F' for false, we can summarize this in a *characteristic truth table* for negation:

$\mathcal{A}$	$\neg\mathcal{A}$
T	F
F	T

We will discuss truth tables at greater length in the next chapter.

Conjunction

Consider these sentences:

9. Adam is athletic.
10. Barbara is athletic.
11. Adam is athletic, and Barbara is also athletic.

We will need separate sentence letters for 9 and 10, so we define this symbolization key:

A: Adam is athletic.
B: Barbara is athletic.

Sentence 9 can be symbolized as A.

Sentence 10 can be symbolized as B.

Sentence 11 can be paraphrased as 'A and B.' In order to fully symbolize this sentence, we need another symbol. We will use '$\wedge$.' We translate 'A and B' as $A \wedge B$. The logical connective '$\wedge$' is called CONJUNCTION, and A and B are each called CONJUNCTS.

Notice that we make no attempt to symbolize 'also' in sentence 11. Words like 'both' and 'also' function to draw our attention to the fact that two things are being conjoined. They are not doing any further logical work, so we do not need to represent them in SL.

Some more examples:

12. Barbara is athletic and energetic.
13. Barbara and Adam are both athletic.
14. Although Barbara is energetic, she is not athletic.
15. Barbara is athletic, but Adam is more athletic than she is.

Sentence 12 is obviously a conjunction. The sentence says two things about Barbara, so in English it is permissible to refer to Barbara only once. It might be tempting to try this when translating the argument: Since B means 'Barbara is athletic', one might paraphrase the sentences as 'B and energetic.' This would be a mistake. Once we translate part of a sentence as B, any further structure is lost. B is an atomic sentence; it is nothing more than true or false. Conversely, 'energetic' is not a sentence; on its own it is neither true nor false. We should instead paraphrase the sentence as 'B and Barbara is energetic.' Now we need to add a sentence letter to the symbolization key. Let E mean 'Barbara is energetic.' Now the sentence can be translated as $B \wedge E$.

> A sentence can be symbolized as $\mathcal{A} \wedge \mathcal{B}$ if it can be paraphrased in English as 'Both $\mathcal{A}$, and $\mathcal{B}$.' Each of the conjuncts must be a sentence.

Sentence 13 says one thing about two different subjects. It says of both Barbara and Adam that they are athletic, and in English we use the word 'athletic' only once. In translating to SL, it is important to realize that the sentence can be paraphrased as, 'Barbara is athletic, and Adam is athletic.' This translates as $B \wedge A$.

Sentence 14 is a bit more complicated. The word 'although' sets up a contrast between the first part of the sentence and the second part. Nevertheless, the sentence says both that Barbara is energetic and that she is not athletic. In order to make each of the conjuncts an atomic sentence, we need to replace 'she' with 'Barbara.'

So we can paraphrase sentence 14 as, '*Both* Barbara is energetic, *and* Barbara is not athletic.' The second conjunct contains a negation, so we paraphrase further: '*Both* Barbara is energetic *and it is not the case that* Barbara is athletic.' This translates as $E \wedge \neg B$.

Sentence 15 contains a similar contrastive structure. It is irrelevant for the purpose of translating to SL, so we can paraphrase the sentence as '*Both* Barbara is athletic, *and* Adam is more athletic than Barbara.' (Notice that we once again replace the pronoun 'she' with her name.) How should we translate the second conjunct? We already have the sentence letter A which is about Adam's being athletic and B which is about Barbara's being athletic, but neither is about one of them being more athletic than the other. We need a new sentence letter. Let R mean 'Adam is more athletic than Barbara.' Now the sentence translates as $B \wedge R$.

> Sentences that can be paraphrased '$\mathcal{A}$, but $\mathcal{B}$' or 'Although $\mathcal{A}$, $\mathcal{B}$' are best symbolized using conjunction: $\mathcal{A} \wedge \mathcal{B}$

It is important to keep in mind that the sentence letters A, B, and R are atomic sentences. Considered as symbols of SL, they have no meaning beyond being true or false. We have used them to symbolize different English language sentences that are all about people being athletic, but this similarity is completely lost when we translate to SL. No formal language can capture all the structure of the English language, but as long as this structure is not important to the argument there is nothing lost by leaving it out.

For any sentences $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \wedge \mathcal{B}$ is true if and only if both $\mathcal{A}$ and $\mathcal{B}$ are true. We can summarize this in the characteristic truth table for conjunction:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\wedge\mathcal{B}$
T	T	T
T	F	F
F	T	F
F	F	F

Conjunction is *symmetrical* because we can swap the conjuncts without changing the truth-value of the sentence. Regardless of what $\mathcal{A}$ and $\mathcal{B}$ are, $\mathcal{A}\wedge\mathcal{B}$ is logically equivalent to $\mathcal{B}\wedge\mathcal{A}$.

Disjunction

Consider these sentences:

16. Either Denison will play golf with me, or he will watch movies.
17. Either Denison or Ellery will play golf with me.

For these sentences we can use this symbolization key:

D: Denison will play golf with me.
E: Ellery will play golf with me.
M: Denison will watch movies.

Sentence 16 is 'Either D or M.' To fully symbolize this, we introduce a new symbol. The sentence becomes $D \vee M$. The '$\vee$' connective is called DISJUNCTION, and D and M are called DISJUNCTS.

Sentence 17 is only slightly more complicated. There are two subjects, but the English sentence only gives the verb once. In translating, we can paraphrase it as. 'Either Denison will play golf with me, or Ellery will play golf with me.' Now it obviously translates as $D \vee E$.

A sentence can be symbolized as $\mathcal{A} \vee \mathcal{B}$ if it can be paraphrased in English as 'Either $\mathcal{A}$, or $\mathcal{B}$.' Each of the disjuncts must be a sentence.

Sometimes in English, the word 'or' excludes the possibility that both disjuncts are true. This is called an EXCLUSIVE OR. An *exclusive or* is clearly intended when it says, on a restaurant menu, 'Entrees come with either soup or salad.' You may have soup; you may have salad; but, if you want *both* soup *and* salad, then you have to pay extra.

At other times, the word 'or' allows for the possibility that both disjuncts might be true. This is probably the case with sentence 17, above. I might play with

Denison, with Ellery, or with both Denison and Ellery. Sentence 17 merely says that I will play with *at least* one of them. This is called an INCLUSIVE OR.

The symbol '$\vee$' represents an *inclusive or*. So $D \vee E$ is true if D is true, if E is true, or if both D and E are true. It is false only if both D and E are false. We can summarize this with the characteristic truth table for disjunction:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\vee\mathcal{B}$
T	T	T
T	F	T
F	T	T
F	F	F

Like conjunction, disjunction is symmetrical. $\mathcal{A}\vee\mathcal{B}$ is logically equivalent to $\mathcal{B}\vee\mathcal{A}$.

These sentences are somewhat more complicated:

18. Either you will not have soup, or you will not have salad.
19. You will have neither soup nor salad.
20. You get either soup or salad, but not both.

We let S_1 mean that you get soup and S_2 mean that you get salad.

Sentence 18 can be paraphrased in this way: 'Either *it is not the case that* you get soup, or *it is not the case that* you get salad.' Translating this requires both disjunction and negation. It becomes $\neg S_1 \vee \neg S_2$.

Sentence 19 also requires negation. It can be paraphrased as, '*It is not the case that* either that you get soup or that you get salad.' We need some way of indicating that the negation does not just negate the right or left disjunct, but rather negates the entire disjunction. In order to do this, we put parentheses around the disjunction: 'It is not the case that $(S_1 \vee S_2)$.' This becomes simply $\neg(S_1 \vee S_2)$.

Notice that the parentheses are doing important work here. The sentence $\neg S_1 \vee S_2$ would mean 'Either you will not have soup, or you will have salad.'

Sentence 20 is an *exclusive or*. We can break the sentence into two parts. The first part says that you get one or the other. We translate this as $(S_1 \vee S_2)$. The second part says that you do not get both. We can paraphrase this as, 'It is not the case both that you get soup and that you get salad.' Using both negation and conjunction, we translate this as $\neg(S_1 \wedge S_2)$. Now we just need to put the two parts together. As we saw above, 'but' can usually be translated as a conjunction. Sentence 20 can thus be translated as $(S_1 \vee S_2) \wedge \neg(S_1 \wedge S_2)$.

Although '$\vee$' is an *inclusive or*, we can symbolize an *exclusive or* in SL. We just need more than one connective to do it.

Conditional

For the following sentences, let R mean 'You will cut the red wire' and B mean 'The bomb will explode.'

21. If you cut the red wire, then the bomb will explode.
22. The bomb will explode only if you cut the red wire.

Sentence 21 can be translated partially as 'If R, then B.' We will use the symbol '$\rightarrow$' to represent logical entailment. The sentence becomes $R \rightarrow B$. The connective is called a CONDITIONAL. The sentence on the left-hand side of the conditional (R in this example) is called the ANTECEDENT. The sentence on the right-hand side (B) is called the CONSEQUENT.

Sentence 22 is also a conditional. Since the word 'if' appears in the second half of the sentence, it might be tempting to symbolize this in the same way as sentence 21. That would be a mistake.

The conditional $R \rightarrow B$ says that *if* R were true, *then* B would also be true. It does not say that your cutting the red wire is the *only* way that the bomb could explode. Someone else might cut the wire, or the bomb might be on a timer. The sentence $R \rightarrow B$ does not say anything about what to expect if R is false. Sentence 22 is different. It says that the only conditions under which the bomb will explode involve your having cut the red wire; i.e., if the bomb explodes, then you must have cut the wire. As such, sentence 22 should be symbolized as $B \rightarrow R$.

It is important to remember that the connective '$\rightarrow$' says only that, if the antecedent is true, then the consequent is true. It says nothing about the *causal* connection between the two events. Translating sentence 22 as $B \rightarrow R$ does not mean that the bomb exploding would somehow have caused your cutting the wire. Both sentence 21 and 22 suggest that, if you cut the red wire, your cutting the red wire would be the cause of the bomb exploding. They differ on the *logical* connection. If sentence 22 were true, then an explosion would tell us— those of us safely away from the bomb— that you had cut the red wire. Without an explosion, sentence 22 tells us nothing.

> The paraphrased sentence '$\mathcal{A}$ only if $\mathcal{B}$' is logically equivalent to 'If $\mathcal{A}$, then $\mathcal{B}$.'

'If $\mathcal{A}$ then $\mathcal{B}$' means that if $\mathcal{A}$ is true then so is $\mathcal{B}$. So we know that if the antecedent $\mathcal{A}$ is true but the consequent $\mathcal{B}$ is false, then the conditional 'If $\mathcal{A}$ then $\mathcal{B}$' is false. What is the truth value of 'If $\mathcal{A}$ then $\mathcal{B}$' under other circumstances? Suppose, for instance, that the antecedent $\mathcal{A}$ happened to be false. 'If $\mathcal{A}$ then $\mathcal{B}$' would then not tell us anything about the actual truth value of the consequent $\mathcal{B}$, and it is unclear what the truth value of 'If $\mathcal{A}$ then $\mathcal{B}$' would be.

In English, the truth of conditionals often depends on what *would* be the case if the antecedent *were true*— even if, as a matter of fact, the antecedent is false. This poses a problem for translating conditionals into SL. Considered as sentences of SL, R and B in the above examples have nothing intrinsic to do with each other. In order to consider what the world would be like if R were true, we would need to analyze what R says about the world. Since R is an atomic symbol of SL, however, there is no further structure to be analyzed. When we replace a sentence with a sentence letter, we consider it merely as some atomic sentence that might be true or false.

In order to translate conditionals into SL, we will not try to capture all the subtleties of the English language 'If... then....' Instead, the symbol '$\rightarrow$' will be a *material conditional*. This means that when $\mathcal{A}$ is false, the conditional $\mathcal{A}\rightarrow\mathcal{B}$ is automatically true, regardless of the truth value of $\mathcal{B}$. If both $\mathcal{A}$ and $\mathcal{B}$ are true, then the conditional $\mathcal{A}\rightarrow\mathcal{B}$ is true.

In short, $\mathcal{A}\rightarrow\mathcal{B}$ is false if and only if $\mathcal{A}$ is true and $\mathcal{B}$ is false. We can summarize this with a characteristic truth table for the conditional.

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\rightarrow\mathcal{B}$
T	T	T
T	F	F
F	T	T
F	F	T

The conditional is *asymmetrical*. You cannot swap the antecedent and consequent without changing the meaning of the sentence, because $\mathcal{A}\rightarrow\mathcal{B}$ and $\mathcal{B}\rightarrow\mathcal{A}$ are not logically equivalent.

Not all sentences of the form 'If... then...' are conditionals. Consider this sentence:

23. If anyone wants to see me, then I will be on the porch.

If I say this, it means that I will be on the porch, regardless of whether anyone wants to see me or not— but if someone did want to see me, then they should look for me there. If we let P mean 'I will be on the porch,' then sentence 23 can be translated simply as P.

Biconditional

Consider these sentences:

24. The figure on the board is a triangle only if it has exactly three sides.
25. The figure on the board is a triangle if it has exactly three sides.
26. The figure on the board is a triangle if and only if it has exactly three sides.

Let T mean 'The figure is a triangle' and S mean 'The figure has three sides.'

Sentence 24, for reasons discussed above, can be translated as $T \rightarrow S$.

Sentence 25 is importantly different. It can be paraphrased as, 'If the figure has three sides, then it is a triangle.' So it can be translated as $S \rightarrow T$.

Sentence 26 says that T is true *if and only if* S is true; we can infer S from T, and we can infer T from S. This is called a BICONDITIONAL, because it entails the two conditionals $S \rightarrow T$ and $T \rightarrow S$. We will use '$\leftrightarrow$' to represent the biconditional; sentence 26 can be translated as $S \leftrightarrow T$.

We could abide without a new symbol for the biconditional. Since sentence 26 means '$T \rightarrow S$ and $S \rightarrow T$,' we could translate it as $(T \rightarrow S) \wedge (S \rightarrow T)$. We would need parentheses to indicate that $(T \rightarrow S)$ and $(S \rightarrow T)$ are separate conjuncts; the expression $T \rightarrow S \wedge S \rightarrow T$ would be ambiguous.

Because we could always write $(\mathcal{A} \rightarrow \mathcal{B}) \wedge (\mathcal{B} \rightarrow \mathcal{A})$ instead of $\mathcal{A} \leftrightarrow \mathcal{B}$, we do not strictly speaking *need* to introduce a new symbol for the biconditional. Nevertheless, logical languages usually have such a symbol. SL will have one, which makes it easier to translate phrases like 'if and only if.'

$\mathcal{A} \leftrightarrow \mathcal{B}$ is true if and only if $\mathcal{A}$ and $\mathcal{B}$ have the same truth value. This is the characteristic truth table for the biconditional:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \leftrightarrow \mathcal{B}$
T	T	T
T	F	F
F	T	F
F	F	T

2.3 Other symbolization

We have now introduced all of the connectives of SL. We can use them together to translate many kinds of sentences. Consider these examples of sentences that use the English-language connective 'unless':

27. Unless you wear a jacket, you will catch cold.
28. You will catch cold unless you wear a jacket.

Let J mean 'You will wear a jacket' and let D mean 'You will catch a cold.'

We can paraphrase sentence 27 as 'Unless J, D.' This means that if you do not wear a jacket, then you will catch cold; with this in mind, we might translate it as $\neg J \rightarrow D$. It also means that if you do not catch a cold, then you must have worn a jacket; with this in mind, we might translate it as $\neg D \rightarrow J$.

Which of these is the correct translation of sentence 27? Both translations are correct, because the two translations are logically equivalent in SL.

Sentence 28, in English, is logically equivalent to sentence 27. It can be translated as either $\neg J \rightarrow D$ or $\neg D \rightarrow J$.

When symbolizing sentences like sentence 27 and sentence 28, it is easy to get turned around. Since the conditional is not symmetric, it would be wrong to translate either sentence as $J \rightarrow \neg D$. Fortunately, there are other logically equivalent expressions. Both sentences mean that you will wear a jacket or— if you do not wear a jacket— then you will catch a cold. So we can translate them as $J \vee D$. (You might worry that the 'or' here should be an *exclusive or*. However, the sentences do not exclude the possibility that you might *both* wear a jacket *and* catch a cold; jackets do not protect you from all the possible ways that you might catch a cold.)

If a sentence can be paraphrased as 'Unless $\mathcal{A}$, $\mathcal{B}$,' then it can be symbolized as $\mathcal{A} \vee \mathcal{B}$.

Symbolization of standard sentence types is summarized on p. 147.

2.4 Sentences of SL

The sentence 'Apples are red, or berries are blue' is a sentence of English, and the sentence '$(A \vee B)$' is a sentence of SL. Although we can identify sentences of English when we encounter them, we do not have a formal definition of 'sentence of English'. In SL, it is possible to formally define what counts as a sentence. This is one respect in which a formal language like SL is more precise than a natural language like English.

It is important to distinguish between the logical language SL, which we are developing, and the language that we use to talk about SL. When we talk about a language, the language that we are talking about is called the OBJECT LANGUAGE. The language that we use to talk about the object language is called the METALANGUAGE.

The object language in this chapter is SL. The metalanguage is English— not conversational English, but English supplemented with some logical and mathematical vocabulary. The sentence '$(A \vee B)$' is a sentence in the object language, because it uses only symbols of SL. The word 'sentence' is not itself part of SL, however, so the sentence 'This expression is a sentence of SL' is not a sentence of SL. It is a sentence in the metalanguage, a sentence that we use to talk *about* SL.

In this section, we will give a formal definition for 'sentence of SL.' The definition itself will be given in mathematical English, the metalanguage.

Expressions

There are three kinds of symbols in SL:

sentence letters with subscripts, as needed	$A, B, C, \ldots, Z$ $A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$
connectives	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
parentheses	(,)

We define an EXPRESSION OF SL as any string of symbols of SL. Take any of the symbols of SL and write them down, in any order, and you have an expression.

Well-formed formulae

Since any sequence of symbols is an expression, many expressions of SL will be gobbledegook. A meaningful expression is called a *well-formed formula*. It is common to use the acronym *wff*; the plural is wffs.

Obviously, individual sentence letters like A and G_{13} will be wffs. We can form further wffs out of these by using the various connectives. Using negation, we can get $\neg A$ and $\neg G_{13}$. Using conjunction, we can get $A \wedge G_{13}$, $G_{13} \wedge A$, $A \wedge A$, and $G_{13} \wedge G_{13}$. We could also apply negation repeatedly to get wffs like $\neg\neg A$ or apply negation along with conjunction to get wffs like $\neg(A \wedge G_{13})$ and $\neg(G_{13} \wedge \neg G_{13})$. The possible combinations are endless, even starting with just these two sentence letters, and there are infinitely many sentence letters. So there is no point in trying to list all the wffs.

Instead, we will describe the process by which wffs can be constructed. Consider negation: Given any wff $\mathcal{A}$ of SL, $\neg\mathcal{A}$ is a wff of SL. It is important here that $\mathcal{A}$ is not the sentence letter A. Rather, it is a variable that stands in for any wff at all. Notice that this variable $\mathcal{A}$ is not a symbol of SL, so $\neg\mathcal{A}$ is not an expression of SL. Instead, it is an expression of the metalanguage that allows us to talk about infinitely many expressions of SL: all of the expressions that start with the negation symbol. Because $\mathcal{A}$ is part of the metalanguage, it is called a *metavariable*.

We can say similar things for each of the other connectives. For instance, if $\mathcal{A}$ and $\mathcal{B}$ are wffs of SL, then $(\mathcal{A} \wedge \mathcal{B})$ is a wff of SL. Providing clauses like this for all of the connectives, we arrive at the following formal definition for a well-formed formula of SL:

1. Every atomic sentence is a wff.
2. If $\mathcal{A}$ is a wff, then $\neg\mathcal{A}$ is a wff of SL.
3. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, then $(\mathcal{A} \wedge \mathcal{B})$ is a wff.
4. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, then $(\mathcal{A} \vee \mathcal{B})$ is a wff.
5. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, then $(\mathcal{A} \rightarrow \mathcal{B})$ is a wff.
6. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, then $(\mathcal{A} \leftrightarrow \mathcal{B})$ is a wff.
7. All and only wffs of SL can be generated by applications of these rules.

Notice that we cannot immediately apply this definition to see whether an arbitrary expression is a wff. Suppose we want to know whether or not $\neg\neg\neg D$ is a wff of SL. Looking at the second clause of the definition, we know that $\neg\neg\neg D$ is a wff *if* $\neg\neg D$ is a wff. So now we need to ask whether or not $\neg\neg D$ is a wff. Again looking at the second clause of the definition, $\neg\neg D$ is a wff *if* $\neg D$ is. Again, $\neg D$ is a wff *if* D is a wff. Now D is a sentence letter, an atomic sentence of SL, so we know that D is a wff by the first clause of the definition. So for a compound formula like $\neg\neg\neg D$, we must apply the definition repeatedly. Eventually we arrive at the atomic sentences from which the wff is built up.

Definitions like this are called *recursive*. Recursive definitions begin with some specifiable base elements and define ways to indefinitely compound the base elements. Just as the recursive definition allows complex sentences to be built up from simple parts, you can use it to decompose sentences into their simpler parts. To determine whether or not something meets the definition, you may have to refer back to the definition many times.

The connective that you look to first in decomposing a sentence is called the MAIN LOGICAL OPERATOR of that sentence. For example: The main logical operator of $\neg(E \vee (F \rightarrow G))$ is negation, $\neg$. The main logical operator of $(\neg E \vee (F \rightarrow G))$ is disjunction, $\vee$.

Sentences

Recall that a sentence is a meaningful expression that can be true or false. Since the meaningful expressions of SL are the wffs and since every wff of SL is either true or false, the definition for a sentence of SL is the same as the definition for a wff. Not every formal language will have this nice feature. In the language QL, which is developed later in the book, there are wffs which are not sentences.

The recursive structure of sentences in SL will be important when we consider the circumstances under which a particular sentence would be true or false. The sentence $\neg\neg\neg D$ is true if and only if the sentence $\neg\neg D$ is false, and so on through the structure of the sentence until we arrive at the atomic components: $\neg\neg\neg D$ is true if and only if the atomic sentence D is false. We will return to this point in the next chapter.

Notational conventions

A wff like $(Q \wedge R)$ must be surrounded by parentheses, because we might apply the definition again to use this as part of a more complicated sentence. If we negate $(Q \wedge R)$, we get $\neg(Q \wedge R)$. If we just had $Q \wedge R$ without the parentheses and put a negation in front of it, we would have $\neg Q \wedge R$. It is most natural to read this as meaning the same thing as $(\neg Q \wedge R)$, something very different than $\neg(Q \wedge R)$. The sentence $\neg(Q \wedge R)$ means that it is not the case that both Q and R are true; Q might be false or R might be false, but the sentence does not tell us which. The sentence $(\neg Q \wedge R)$ means specifically that Q is false and that R is true. As such, parentheses are crucial to the meaning of the sentence.

So, strictly speaking, $Q \wedge R$ without parentheses is *not* a sentence of SL. When using SL, however, we will often be able to relax the precise definition so as to make things easier for ourselves. We will do this in several ways.

First, we understand that $Q \wedge R$ means the same thing as $(Q \wedge R)$. As a matter of convention, we can leave off parentheses that occur *around the entire sentence*.

Second, it can sometimes be confusing to look at long sentences with many, nested pairs of parentheses. We adopt the convention of using square brackets '[' and ']' in place of parenthesis. There is no logical difference between $(P \vee Q)$ and $[P \vee Q]$, for example. The unwieldy sentence

$$(((H \rightarrow I) \vee (I \rightarrow H)) \wedge (J \vee K))$$

could be written in this way:

$$\big[(H \rightarrow I) \vee (I \rightarrow H)\big] \wedge (J \vee K)$$

Third, we will sometimes want to translate the conjunction of three or more sentences. For the sentence 'Alice, Bob, and Candice all went to the party', suppose we let A mean 'Alice went', B mean 'Bob went', and C mean 'Candice went.' The definition only allows us to form a conjunction out of two sentences, so we can translate it as $(A \wedge B) \wedge C$ or as $A \wedge (B \wedge C)$. There is no reason to distinguish between these, since the two translations are logically equivalent. There is no logical difference between the first, in which $(A \wedge B)$ is conjoined with C, and the second, in which A is conjoined with $(B \wedge C)$. So we might as well just write $A \wedge B \wedge C$. As a matter of convention, we can leave out parentheses when we conjoin three or more sentences.

Fourth, a similar situation arises with multiple disjunctions. 'Either Alice, Bob, or Candice went to the party' can be translated as $(A \vee B) \vee C$ or as $A \vee (B \vee C)$. Since these two translations are logically equivalent, we may write $A \vee B \vee C$.

These latter two conventions only apply to multiple conjunctions or multiple disjunctions. If a series of connectives includes both disjunctions and conjunctions, then the parentheses are essential; as with $(A \wedge B) \vee C$ and $A \wedge (B \vee C)$. The parentheses are also required if there is a series of conditionals or biconditionals; as with $(A \rightarrow B) \rightarrow C$ and $A \leftrightarrow (B \leftrightarrow C)$.

We have adopted these four rules as *notational conventions*, not as changes to the definition of a sentence. Strictly speaking, $A \vee B \vee C$ is still not a sentence. Instead, it is a kind of shorthand. We write it for the sake of convenience, but we really mean the sentence $(A \vee (B \vee C))$.

If we had given a different definition for a wff, then these could count as wffs. We might have written rule 3 in this way: "If $\mathcal{A}$, $\mathcal{B}$, ... $\mathcal{Z}$ are wffs, then $(\mathcal{A} \wedge \mathcal{B} \wedge \ldots \wedge \mathcal{Z})$, is a wff." This would make it easier to translate some English sentences, but would have the cost of making our formal language more complicated. We would have to keep the complex definition in mind when we develop truth tables and a proof system. We want a logical language that is *expressively simple* and allows us to translate easily from English, but we also want a *formally simple* language. Adopting notational conventions is a compromise between these two desires.

Practice Exercises

$\star$ **Part A** Using the symbolization key given, translate each English-language sentence into SL.

M: Those creatures are men in suits.
C: Those creatures are chimpanzees.
G: Those creatures are gorillas.

1. Those creatures are not men in suits.
2. Those creatures are men in suits, or they are not.
3. Those creatures are either gorillas or chimpanzees.
4. Those creatures are neither gorillas nor chimpanzees.
5. If those creatures are chimpanzees, then they are neither gorillas nor men in suits.
6. Unless those creatures are men in suits, they are either chimpanzees or they are gorillas.

Part B Using the symbolization key given, translate each English-language sentence into SL.

A: Mister Ace was murdered.
B: The butler did it.
C: The cook did it.
D: The Duchess is lying.
E: Mister Edge was murdered.
F: The murder weapon was a frying pan.

1. Either Mister Ace or Mister Edge was murdered.
2. If Mister Ace was murdered, then the cook did it.
3. If Mister Edge was murdered, then the cook did not do it.
4. Either the butler did it, or the Duchess is lying.
5. The cook did it only if the Duchess is lying.
6. If the murder weapon was a frying pan, then the culprit must have been the cook.
7. If the murder weapon was not a frying pan, then the culprit was either the cook or the butler.
8. Mister Ace was murdered if and only if Mister Edge was not murdered.
9. The Duchess is lying, unless it was Mister Edge who was murdered.
10. If Mister Ace was murdered, he was done in with a frying pan.
11. Since the cook did it, the butler did not.
12. Of course the Duchess is lying!

$\star$ **Part C** Using the symbolization key given, translate each English-language sentence into SL.

$\mathbf{E}_1$**:** Ava is an electrician.
$\mathbf{E}_2$**:** Harrison is an electrician.
$\mathbf{F}_1$**:** Ava is a firefighter.
$\mathbf{F}_2$**:** Harrison is a firefighter.
$\mathbf{S}_1$**:** Ava is satisfied with her career.
$\mathbf{S}_2$**:** Harrison is satisfied with his career.

1. Ava and Harrison are both electricians.
2. If Ava is a firefighter, then she is satisfied with her career.
3. Ava is a firefighter, unless she is an electrician.
4. Harrison is an unsatisfied electrician.
5. Neither Ava nor Harrison is an electrician.
6. Both Ava and Harrison are electricians, but neither of them find it satisfying.
7. Harrison is satisfied only if he is a firefighter.
8. If Ava is not an electrician, then neither is Harrison, but if she is, then he is too.
9. Ava is satisfied with her career if and only if Harrison is not satisfied with his.

10. If Harrison is both an electrician and a firefighter, then he must be satisfied with his work.
11. It cannot be that Harrison is both an electrician and a firefighter.
12. Harrison and Ava are both firefighters if and only if neither of them is an electrician.

⋆ **Part D** Give a symbolization key and symbolize the following sentences in SL.

1. Alice and Bob are both spies.
2. If either Alice or Bob is a spy, then the code has been broken.
3. If neither Alice nor Bob is a spy, then the code remains unbroken.
4. The German embassy will be in an uproar, unless someone has broken the code.
5. Either the code has been broken or it has not, but the German embassy will be in an uproar regardless.
6. Either Alice or Bob is a spy, but not both.

Part E Give a symbolization key and symbolize the following sentences in SL.

1. If Gregor plays first base, then the team will lose.
2. The team will lose unless there is a miracle.
3. The team will either lose or it won't, but Gregor will play first base regardless.
4. Gregor's mom will bake cookies if and only if Gregor plays first base.
5. If there is a miracle, then Gregor's mom will not bake cookies.

Part F For each argument, write a symbolization key and translate the argument as well as possible into SL.

1. If Dorothy plays the piano in the morning, then Roger wakes up cranky. Dorothy plays piano in the morning unless she is distracted. So if Roger does not wake up cranky, then Dorothy must be distracted.
2. It will either rain or snow. If it rains, Neville will be sad. If it snows, Neville will be cold. Therefore, Neville will either be sad or cold.
3. If Zoog remembered to do his chores, then things are clean but not neat. If he forgot, then things are neat but not clean. Therefore, things are either neat or clean— but not both.

⋆ **Part G** For each of the following: (a) Is it a wff of SL? (b) Is it a sentence of SL, allowing for notational conventions?

1. (A)
2. $J_{374} \vee \neg J_{374}$

3. $\neg\neg\neg\neg F$
4. $\neg \wedge S$
5. $(G \wedge \neg G)$
6. $\mathcal{A} \rightarrow \mathcal{A}$
7. $(A \rightarrow (A \wedge \neg F)) \vee (D \leftrightarrow E)$
8. $[(Z \leftrightarrow S) \rightarrow W] \wedge [J \vee X]$
9. $(F \leftrightarrow \neg D \rightarrow J) \vee (C \wedge D)$

Part H

1. Are there any wffs of SL that contain no sentence letters? Why or why not?
2. In the chapter, we symbolized an *exclusive or* using $\vee$, $\wedge$, and $\neg$. How could you translate an *exclusive or* using only two connectives? Is there any way to translate an *exclusive or* using only one connective?

Chapter 3

Truth tables

This chapter introduces a way of evaluating sentences and arguments of SL. Although it can be laborious, the truth table method is a purely mechanical procedure that requires no intuition or special insight.

3.1 Truth-functional connectives

Any non-atomic sentence of SL is composed of atomic sentences with sentential connectives. The truth-value of the compound sentence depends only on the truth-value of the atomic sentences that comprise it. In order to know the truth-value of $(D \leftrightarrow E)$, for instance, you only need to know the truth-value of D and the truth-value of E. Connectives that work in this way are called TRUTH-FUNCTIONAL.

In this chapter, we will make use of the fact that all of the logical operators in SL are truth-functional— it makes it possible to construct truth tables to determine the logical features of sentences. You should realize, however, that this is not possible for all languages. In English, it is possible to form a new sentence from any simpler sentence $\mathcal{X}$ by saying 'It is possible that $\mathcal{X}$.' The truth-value of this new sentence does not depend directly on the truth-value of $\mathcal{X}$. Even if $\mathcal{X}$ is false, perhaps in some sense $\mathcal{X}$ *could* have been true— then the new sentence would be true. Some formal languages, called *modal logics*, have an operator for possibility. In a modal logic, we could translate 'It is possible that $\mathcal{X}$' as $\diamond\mathcal{X}$. However, the ability to translate sentences like these come at a cost: The $\diamond$ operator is not truth-functional, and so modal logics are not amenable to truth tables.

3.2 Complete truth tables

The truth-value of sentences which contain only one connective are given by the characteristic truth table for that connective. In the previous chapter, we wrote the characteristic truth tables with 'T' for true and 'F' for false. It is important to note, however, that this is not about truth in any deep or cosmic sense. Poets and philosophers can argue at length about the nature and significance *truth*, but the truth functions in SL are just rules which transform input values into output values. To underscore this, in this chapter we will write '1' and '0' instead of 'T' and 'F'. Even though we interpret '1' as meaning 'true' and '0' as meaning 'false', computers can be programmed to fill out truth tables in a purely mechanical way. In a machine, '1' might mean that a register is switched on and '0' that the register is switched off. Mathematically, they are just the two possible values that a sentence of SL can have.

Here are the truth tables for the connectives of SL, written in terms of 1s and 0s.

$\mathcal{A}$	$\neg\mathcal{A}$
1	0
0	1

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\wedge\mathcal{B}$	$\mathcal{A}\vee\mathcal{B}$	$\mathcal{A}\rightarrow\mathcal{B}$	$\mathcal{A}\leftrightarrow\mathcal{B}$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	0	1	1	0
0	0	0	0	1	1

Table 3.1: The characteristic truth tables for the connectives of SL.

The characteristic truth table for conjunction, for example, gives the truth conditions for any sentence of the form $(\mathcal{A} \wedge \mathcal{B})$. Even if the conjuncts $\mathcal{A}$ and $\mathcal{B}$ are long, complicated sentences, the conjunction is true if and only if both $\mathcal{A}$ and $\mathcal{B}$ are true. Consider the sentence $(H \wedge I) \rightarrow H$. We consider all the possible combinations of true and false for H and I, which gives us four rows. We then copy the truth-values for the sentence letters and write them underneath the letters in the sentence.

H	I	$(H$	$\wedge$	$I)$	$\rightarrow$	H
1	1	**1**		**1**		**1**
1	0	**1**		**0**		**1**
0	1	**0**		**1**		**0**
0	0	**0**		**0**		**0**

Now consider the subsentence $H \wedge I$. This is a conjunction $\mathcal{A}\wedge\mathcal{B}$ with H as $\mathcal{A}$ and with I as $\mathcal{B}$. H and I are both true on the first row. Since a conjunction is true when both conjuncts are true, we write a 1 underneath the conjunction symbol. We continue for the other three rows and get this:

H	I	$(H$	$\wedge$	$I)$	$\rightarrow$	H
		$\mathcal{A}$	$\wedge$	$\mathcal{B}$		
1	1	1	**1**	1		1
1	0	1	**0**	0		1
0	1	0	**0**	1		0
0	0	0	**0**	0		0

The entire sentence is a conditional $\mathcal{A} \rightarrow \mathcal{B}$ with $(H \wedge I)$ as $\mathcal{A}$ and with H as $\mathcal{B}$. On the second row, for example, $(H \wedge I)$ is false and H is true. Since a conditional is true when the antecedent is false, we write a 1 in the second row underneath the conditional symbol. We continue for the other three rows and get this:

H	I	$(H$	$\wedge$	$I)$	$\rightarrow$	H
			$\mathcal{A}$		$\rightarrow$	$\mathcal{B}$
1	1		1		**1**	1
1	0		0		**1**	1
0	1		0		**1**	0
0	0		0		**1**	0

The column of 1s underneath the conditional tells us that the sentence $(H \wedge I) \rightarrow I$ is true regardless of the truth-values of H and I. They can be true or false in any combination, and the compound sentence still comes out true. It is crucial that we have considered all of the possible combinations. If we only had a two-line truth table, we could not be sure that the sentence was not false for some other combination of truth-values.

In this example, we have not repeated all of the entries in every successive table. When actually writing truth tables on paper, however, it is impractical to erase whole columns or rewrite the whole table for every step. Although it is more crowded, the truth table can be written in this way:

H	I	$(H$	$\wedge$	$I)$	$\rightarrow$	H
1	1	1	1	1	1	1
1	0	1	0	0	1	1
0	1	0	0	1	1	0
0	0	0	0	0	1	0

Most of the columns underneath the sentence are only there for bookkeeping purposes. When you become more adept with truth tables, you will probably no longer need to copy over the columns for each of the sentence letters. In any case, the truth-value of the sentence on each row is just the column underneath the main logical operator of the sentence; in this case, the column underneath the conditional.

A COMPLETE TRUTH TABLE has a row for all the possible combinations of 1 and 0 for all of the sentence letters. The size of the complete truth table depends on the number of different sentence letters in the table. A sentence that contains only one sentence letter requires only two rows, as in the characteristic truth table for negation. This is true even if the same letter is repeated many times, as in the sentence $[(C \leftrightarrow C) \rightarrow C] \wedge \neg(C \rightarrow C)$. The complete truth table requires only two lines because there are only two possibilities: C can be true or it can be false. A single sentence letter can never be marked both 1 and 0 on the same row. The truth table for this sentence looks like this:

C	$[(C \leftrightarrow C) \rightarrow C] \wedge \neg(C \rightarrow C)$
1	1 1 1 1 1 **0** 0 1 1 1
0	0 1 0 0 0 **0** 0 0 1 0

Looking at the column underneath the main connective, we see that the sentence is false on both rows of the table; i.e., it is false regardless of whether C is true or false.

A sentence that contains two sentence letters requires four lines for a complete truth table, as in the characteristic truth tables and the table for $(H \wedge I) \rightarrow I$.

A sentence that contains three sentence letters requires eight lines. For example:

M	N	P	$M \wedge (N \vee P)$
1	1	1	1 **1** 1 1 1
1	1	0	1 **1** 1 1 0
1	0	1	1 **1** 0 1 1
1	0	0	1 **0** 0 0 0
0	1	1	0 **0** 1 1 1
0	1	0	0 **0** 1 1 0
0	0	1	0 **0** 0 1 1
0	0	0	0 **0** 0 0 0

From this table, we know that the sentence $M \wedge (N \vee P)$ might be true or false, depending on the truth-values of M, N, and P.

A complete truth table for a sentence that contains four different sentence letters requires 16 lines. Five letters, 32 lines. Six letters, 64 lines. And so on. To be perfectly general: If a complete truth table has n different sentence letters, then it must have 2^n rows.

In order to fill in the columns of a complete truth table, begin with the rightmost sentence letter and alternate 1s and 0s. In the next column to the left, write two 1s, write two 0s, and repeat. For the third sentence letter, write four 1s followed by four 0s. This yields an eight line truth table like the one above. For a 16 line truth table, the next column of sentence letters should have eight 1s followed by eight 0s. For a 32 line table, the next column would have 16 1s followed by 16 0s. And so on.

3.3 Using truth tables

Tautologies, contradictions, and contingent sentences

Recall that an English sentence is a tautology if it must be true as a matter of logic. With a complete truth table, we consider all of the ways that the world might be. If the sentence is true on every line of a complete truth table, then it is true as a matter of logic, regardless of what the world is like.

So a sentence is a TAUTOLOGY IN SL if the column under its main connective is 1 on every row of a complete truth table.

Conversely, a sentence is a CONTRADICTION IN SL if the column under its main connective is 0 on every row of a complete truth table.

A sentence is CONTINGENT IN SL if it is neither a tautology nor a contradiction; i.e. if it is 1 on at least one row and 0 on at least one row.

From the truth tables in the previous section, we know that $(H \wedge I) \rightarrow H$ is a tautology, that $[(C \leftrightarrow C) \rightarrow C] \wedge \neg(C \rightarrow C)$ is a contradiction, and that $M \wedge (N \vee P)$ is contingent.

Logical equivalence

Two sentences are logically equivalent in English if they have the same truth value as a matter logic. Once again, truth tables allow us to define an analogous concept for SL: Two sentences are LOGICALLY EQUIVALENT IN SL if they have the same truth-value on every row of a complete truth table.

Consider the sentences $\neg(A \vee B)$ and $\neg A \wedge \neg B$. Are they logically equivalent? To find out, we construct a truth table.

A	B	$\neg\,(A \vee B)$	$\neg\,A \wedge \neg\,B$
1	1	**0** 1 1 1	0 1 **0** 0 1
1	0	**0** 1 1 0	0 1 **0** 1 0
0	1	**0** 0 1 1	1 0 **0** 0 1
0	0	**1** 0 0 0	1 0 **1** 1 0

Look at the columns for the main connectives; negation for the first sentence, conjunction for the second. On the first three rows, both are 0. On the final row, both are 1. Since they match on every row, the two sentences are logically equivalent.

Consistency

A set of sentences in English is consistent if it is logically possible for them all to be true at once. A set of sentences is LOGICALLY CONSISTENT IN SL if there is at least one line of a complete truth table on which all of the sentences are true. It is INCONSISTENT otherwise.

Validity

An argument in English is valid if it is logically impossible for the premises to be true and for the conclusion to be false at the same time. An argument is VALID IN SL if there is no row of a complete truth table on which the premises are all 1 and the conclusion is 0; an argument is INVALID IN SL if there is such a row.

Consider this argument:

$$\begin{array}{rl} & \neg L \rightarrow (J \vee L) \\ & \neg L \\ \therefore & J \end{array}$$

Is it valid? To find out, we construct a truth table.

J	L	$\neg\, L \rightarrow (J \vee L)$	$\neg$ L	J
1	1	0 1 **1** 1 1 1	**0** 1	**1**
1	0	1 0 **1** 1 1 0	**1** 0	**1**
0	1	0 1 **1** 0 1 1	**0** 1	**0**
0	0	1 0 **0** 0 0 0	**1** 0	**0**

Yes, the argument is valid. The only row on which both the premises are 1 is the second row, and on that row the conclusion is also 1.

3.4 Partial truth tables

In order to show that a sentence is a tautology, we need to show that it is 1 on every row. So we need a complete truth table. To show that a sentence is *not* a tautology, however, we only need one line: a line on which the sentence is 0. Therefore, in order to show that something is not a tautology, it is enough to provide a one-line *partial truth table*— regardless of how many sentence letters the sentence might have in it.

Consider, for example, the sentence $(U \wedge T) \rightarrow (S \wedge W)$. We want to show that it is *not* a tautology by providing a partial truth table. We fill in 0 for the entire sentence. The main connective of the sentence is a conditional. In order for the conditional to be false, the antecedent must be true (1) and the consequent must be false (0). So we fill these in on the table:

S	T	U	W	$(U \wedge T) \rightarrow (S \wedge W)$
				1 **0** 0

In order for the $(U \wedge T)$ to be true, both U and T must be true.

S	T	U	W	$(U \wedge T) \rightarrow (S \wedge W)$
	1	1		1 1 1 **0** 0

Now we just need to make $(S \wedge W)$ false. To do this, we need to make at least one of S and W false. We can make both S and W false if we want. All that matters is that the whole sentence turns out false on this line. Making an arbitrary decision, we finish the table in this way:

S	T	U	W	$(U \wedge T) \rightarrow (S \wedge W)$
0	1	1	0	1 1 1 **0** 0 0 0

Showing that something is a contradiction requires a complete truth table. Showing that something is *not* a contradiction requires only a one-line partial truth table, where the sentence is true on that one line.

A sentence is contingent if it is neither a tautology nor a contradiction. So showing that a sentence is contingent requires a *two-line* partial truth table: The sentence must be true on one line and false on the other. For example, we can show that the sentence above is contingent with this truth table:

S	T	U	W	$(U \wedge T) \rightarrow (S \wedge W)$
0	1	1	0	1 1 1 **0** 0 0 0
0	1	0	0	0 0 1 **1** 0 0 0

Note that there are many combinations of truth values that would have made the sentence true, so there are many ways we could have written the second line.

Showing that a sentence is *not* contingent requires providing a complete truth table, because it requires showing that the sentence is a tautology or that it is a contradiction. If you do not know whether a particular sentence is contingent, then you do not know whether you will need a complete or partial truth table. You can always start working on a complete truth table. If you complete rows that show the sentence is contingent, then you can stop. If not, then complete the truth table. Even though two carefully selected rows will show that a contingent sentence is contingent, there is nothing wrong with filling in more rows.

Showing that two sentences are logically equivalent requires providing a complete truth table. Showing that two sentences are *not* logically equivalent requires only a one-line partial truth table: Make the table so that one sentence is true and the other false.

Showing that a set of sentences is consistent requires providing one row of a truth table on which all of the sentences are true. The rest of the table is irrelevant, so a one-line partial truth table will do. Showing that a set of sentences is inconsistent, on the other hand, requires a complete truth table: You must show that on every row of the table at least one of the sentences is false.

Showing that an argument is valid requires a complete truth table. Showing that an argument is *invalid* only requires providing a one-line truth table: If you can produce a line on which the premises are all true and the conclusion is false, then the argument is invalid.

Here is a table that summarizes when a complete truth table is required and when a partial truth table will do.

	YES	NO
tautology?	complete truth table	one-line partial truth table
contradiction?	complete truth table	one-line partial truth table
contingent?	two-line partial truth table	complete truth table
equivalent?	complete truth table	one-line partial truth table
consistent?	one-line partial truth table	complete truth table
valid?	complete truth table	one-line partial truth table

Table 3.2: Do you need a complete truth table or a partial truth table? It depends on what you are trying to show.

Practice Exercises

If you want additional practice, you can construct truth tables for any of the sentences and arguments in the exercises for the previous chapter.

⋆ **Part A** Determine whether each sentence is a tautology, a contradiction, or a contingent sentence. Justify your answer with a complete or partial truth table where appropriate.

1. $A \to A$
2. $\neg B \wedge B$
3. $C \to \neg C$
4. $\neg D \vee D$
5. $(A \leftrightarrow B) \leftrightarrow \neg(A \leftrightarrow \neg B)$

6. $(A \wedge B) \vee (B \wedge A)$
7. $(A \rightarrow B) \vee (B \rightarrow A)$
8. $\neg[A \rightarrow (B \rightarrow A)]$
9. $(A \wedge B) \rightarrow (B \vee A)$
10. $A \leftrightarrow [A \rightarrow (B \wedge \neg B)]$
11. $\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$
12. $\neg(A \wedge B) \leftrightarrow A$
13. $\big[(A \wedge B) \wedge \neg(A \wedge B)\big] \wedge C$
14. $A \rightarrow (B \vee C)$
15. $[(A \wedge B) \wedge C] \rightarrow B$
16. $(A \wedge \neg A) \rightarrow (B \vee C)$
17. $\neg\big[(C \vee A) \vee B\big]$
18. $(B \wedge D) \leftrightarrow [A \leftrightarrow (A \vee C)]$

⋆ **Part B** Determine whether each pair of sentences is logically equivalent. Justify your answer with a complete or partial truth table where appropriate.

1. A, $\neg A$
2. A, $A \vee A$
3. $A \rightarrow A$, $A \leftrightarrow A$
4. $A \vee \neg B$, $A \rightarrow B$
5. $A \wedge \neg A$, $\neg B \leftrightarrow B$
6. $\neg(A \wedge B)$, $\neg A \vee \neg B$
7. $\neg(A \rightarrow B)$, $\neg A \rightarrow \neg B$
8. $(A \rightarrow B)$, $(\neg B \rightarrow \neg A)$
9. $[(A \vee B) \vee C]$, $[A \vee (B \vee C)]$
10. $[(A \vee B) \wedge C]$, $[A \vee (B \wedge C)]$

⋆ **Part C** Determine whether each set of sentences is consistent or inconsistent. Justify your answer with a complete or partial truth table where appropriate.

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

⋆ **Part D** Determine whether each argument is valid or invalid. Justify your answer with a complete or partial truth table where appropriate.

1. $A \rightarrow A$, $\therefore A$
2. $A \vee \big[A \rightarrow (A \leftrightarrow A)\big]$, $\therefore$ A
3. $A \rightarrow (A \wedge \neg A)$, $\therefore \neg A$

4. $A \leftrightarrow \neg(B \leftrightarrow A)$, $\therefore A$
5. $A \vee (B \rightarrow A)$, $\therefore \neg A \rightarrow \neg B$
6. $A \rightarrow B$, B, $\therefore A$
7. $A \vee B$, $B \vee C$, $\neg A$, $\therefore B \wedge C$
8. $A \vee B$, $B \vee C$, $\neg B$, $\therefore A \wedge C$
9. $(B \wedge A) \rightarrow C$, $(C \wedge A) \rightarrow B$, $\therefore (C \wedge B) \rightarrow A$
10. $A \leftrightarrow B$, $B \leftrightarrow C$, $\therefore A \leftrightarrow C$

⋆ **Part E** Answer each of the questions below and justify your answer.

1. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are logically equivalent. What can you say about $\mathcal{A} \leftrightarrow \mathcal{B}$?
2. Suppose that $(\mathcal{A} \wedge \mathcal{B}) \rightarrow \mathcal{C}$ is contingent. What can you say about the argument "$\mathcal{A}$, $\mathcal{B}$, $\therefore \mathcal{C}$"?
3. Suppose that $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ is inconsistent. What can you say about $(\mathcal{A} \wedge \mathcal{B} \wedge \mathcal{C})$?
4. Suppose that $\mathcal{A}$ is a contradiction. What can you say about the argument "$\mathcal{A}$, $\mathcal{B}$, $\therefore \mathcal{C}$"?
5. Suppose that $\mathcal{C}$ is a tautology. What can you say about the argument "$\mathcal{A}$, $\mathcal{B}$, $\therefore \mathcal{C}$"?
6. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are logically equivalent. What can you say about $(\mathcal{A} \vee \mathcal{B})$?
7. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are *not* logically equivalent. What can you say about $(\mathcal{A} \vee \mathcal{B})$?

Part F We could leave the biconditional ($\leftrightarrow$) out of the language. If we did that, we could still write '$A \leftrightarrow B$' so as to make sentences easier to read, but that would be shorthand for $(A \rightarrow B) \wedge (B \rightarrow A)$. The resulting language would be formally equivalent to SL, since $A \leftrightarrow B$ and $(A \rightarrow B) \wedge (B \rightarrow A)$ are logically equivalent in SL. If we valued formal simplicity over expressive richness, we could replace more of the connectives with notational conventions and still have a language equivalent to SL.

There are a number of equivalent languages with only two connectives. It would be enough to have only negation and the material conditional. Show this by writing sentences that are logically equivalent to each of the following using only parentheses, sentence letters, negation ($\neg$), and the material conditional ($\rightarrow$).

⋆ 1. $A \vee B$
⋆ 2. $A \wedge B$
⋆ 3. $A \leftrightarrow B$

We could have a language that is equivalent to SL with only negation and disjunction as connectives. Show this: Using only parentheses, sentence letters, negation ($\neg$), and disjunction ($\vee$), write sentences that are logically equivalent to each of the following.

4. $A \wedge B$
5. $A \rightarrow B$
6. $A \leftrightarrow B$

The *Sheffer stroke* is a logical connective with the following characteristic truthtable:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\vert\mathcal{B}$
1	1	0
1	0	1
0	1	1
0	0	1

7. Write a sentence using the connectives of SL that is logically equivalent to $(A|B)$.

Every sentence written using a connective of SL can be rewritten as a logically equivalent sentence using one or more Sheffer strokes. Using only the Sheffer stroke, write sentences that are equivalent to each of the following.

8. $\neg A$
9. $(A \wedge B)$
10. $(A \vee B)$
11. $(A \rightarrow B)$
12. $(A \leftrightarrow B)$

Chapter 4

Proofs in SL

Consider two arguments in SL:

Argument A

$$
\begin{array}{ll}
 & P \vee Q \\
 & \neg P \\
\therefore & \text{Q}
\end{array}
$$

Argument B

$$
\begin{array}{ll}
 & P \rightarrow Q \\
 & P \\
\therefore & \text{Q}
\end{array}
$$

Clearly, these are valid arguments. You can confirm that they are valid by constructing four-line truth tables. Argument A makes use of an inference form that is always valid: Given a disjunction and the negation of one of the disjuncts, the other disjunct follows as a valid consequence. This rule is called *disjunctive syllogism.*

Argument B makes use of a different valid form: Given a conditional and its antecedent, the consequent follows as a valid consequence. This is called *modus ponens.*

When we construct truth tables, we do not need to give names to different inference forms. There is no reason to distinguish modus ponens from a disjunctive syllogism. For this same reason, however, the method of truth tables does not clearly show *why* an argument is valid. If you were to do a 1024-line truth table for an argument that contains ten sentence letters, then you could check to see if there were any lines on which the premises were all true and the conclusion were false. If you did not see such a line and provided you made no mistakes in constructing the table, then you would know that the argument was valid. Yet you would not be able to say anything further about why this particular argument was a valid argument form.

The aim of a *proof system* is to show that particular arguments are valid in a way that allows us to understand the reasoning involved in the argument. We begin with basic argument forms, like disjunctive syllogism and modus ponens. These forms can then be combined to make more complicated arguments, like this one:

(1) $\neg L \rightarrow (J \vee L)$
(2) $\neg L$
$\therefore$ J

By modus ponens, (1) and (2) entail $J \vee L$. This is an *intermediate conclusion.* It follows logically from the premises, but it is not the conclusion we want. Now $J \vee L$ and (2) entail J, by disjunctive syllogism. We do not need a new rule for this argument. The proof of the argument shows that it is really just a combination of rules we have already introduced.

Formally, a PROOF is a sequence of sentences. The first sentences of the sequence are assumptions; these are the premises of the argument. Every sentence later in the sequence follows from earlier sentences by one of the rules of proof. The final sentence of the sequence is the conclusion of the argument.

This chapter begins with a proof system for SL, which is then extended to cover QL and QL plus identity.

4.1 Basic rules for SL

In designing a proof system, we could just start with disjunctive syllogism and modus ponens. Whenever we discovered a valid argument which could not be proven with rules we already had, we could introduce new rules. Proceeding in this way, we would have an unsystematic grab bag of rules. We might accidently add some strange rules, and we would surely end up with more rules than we need.

Instead, we will develop what is called a NATURAL DEDUCTION system. In a natural deduction system, there will be two rules for each logical operator: an INTRODUCTION rule that allows us to prove a sentence that has it as the main logical operator and an ELIMINATION rule that allows us to prove something given a sentence that has it as the main logical operator.

In addition to the rules for each logical operator, we will also have a reiteration rule. If you already have shown something in the course of a proof, the reiteration rule allows you to repeat it on a new line. For instance:

1	$\mathcal{A}$	
2	$\mathcal{A}$	R 1

When we add a line to a proof, we write the rule that justifies that line. We also write the numbers of the lines to which the rule was applied. The reiteration rule above is justified by one line, the line that you are reiterating. So the 'R 1' on line 2 of the proof means that the line is justified by the reiteration rule (R) applied to line 1.

Obviously, the reiteration rule will not allow us to show anything *new*. For that, we will need more rules. The remainder of this section will give introduction and elimination rules for all of the sentential connectives. This will give us a complete proof system for SL. Later in the chapter, we introduce rules for quantifiers and identity.

All of the rules introduced in this chapter are summarized starting on p. 150.

Conjunction

Think for a moment: What would you need to show in order to prove $E \wedge F$?

Of course, you could show $E \wedge F$ by proving E and separately proving F. This holds even if the two conjuncts are not atomic sentences. If you can prove $[(A \vee J) \rightarrow V]$ and $[(V \rightarrow L) \leftrightarrow (F \vee N)]$, then you have effectively proven

$$[(A \vee J) \rightarrow V] \wedge [(V \rightarrow L) \leftrightarrow (F \vee N)].$$

So this will be our conjunction introduction rule, which we abbreviate $\wedge$I:

m	$\mathcal{A}$	
n	$\mathcal{B}$	
	$\mathcal{A} \wedge \mathcal{B}$	$\wedge$I m, n

A line of proof must be justified by some rule, and here we have '$\wedge$I m,n.' This means: Conjunction introduction applied to line m and line n. These are variables, not real line numbers; m is some line and n is some other line. In an actual proof, the lines are numbered $1, 2, 3, \ldots$ and rules must be applied to specific line numbers. When we define the rule, however, we use variables to underscore the point that the rule may be applied to any two lines that are already in the proof. If you have K on line 8 and L on line 15, you can prove $(K \wedge L)$ at some later point in the proof with the justification '$\wedge$I 8, 15.'

Now, consider the elimination rule for conjunction. What are you entitled to conclude from a sentence like $E \wedge F$? Surely, you are entitled to conclude E; if $E \wedge F$ were true, then E would be true. Similarly, you are entitled to conclude F. This will be our conjunction elimination rule, which we abbreviate $\wedge$E:

m	$\mathcal{A} \wedge \mathcal{B}$	
	$\mathcal{A}$	$\wedge$E m
	$\mathcal{B}$	$\wedge$E m

When you have a conjunction on some line of a proof, you can use $\wedge$E to derive either of the conjuncts. The $\wedge$E rule requires only one sentence, so we write one line number as the justification for applying it.

Even with just these two rules, we can provide some proofs. Consider this argument.

$$[(A \vee B) \to (C \vee D)] \wedge [(E \vee F) \to (G \vee H)]$$
$$\therefore\ [(E \vee F) \to (G \vee H)] \wedge [(A \vee B) \to (C \vee D)]$$

The main logical operator in both the premise and conclusion is conjunction. Since conjunction is symmetric, the argument is obviously valid. In order to provide a proof, we begin by writing down the premise. After the premises, we draw a horizontal line— everything below this line must be justified by a rule of proof. So the beginning of the proof looks like this:

1	$[(A \vee B) \to (C \vee D)] \wedge [(E \vee F) \to (G \vee H)]$

From the premise, we can get each of the conjuncts by $\wedge$E. The proof now looks like this:

1	$[(A \vee B) \to (C \vee D)] \wedge [(E \vee F) \to (G \vee H)]$	
2	$[(A \vee B) \to (C \vee D)]$	$\wedge$E 1
3	$[(E \vee F) \to (G \vee H)]$	$\wedge$E 1

The rule $\wedge$I requires that we have each of the conjuncts available somewhere in the proof. They can be separated from one another, and they can appear in any order. So by applying the $\wedge$I rule to lines 3 and 2, we arrive at the desired conclusion. The finished proof looks like this:

1	$[(A \vee B) \to (C \vee D)] \wedge [(E \vee F) \to (G \vee H)]$	
2	$[(A \vee B) \to (C \vee D)]$	$\wedge$E 1
3	$[(E \vee F) \to (G \vee H)]$	$\wedge$E 1
4	$[(E \vee F) \to (G \vee H)] \wedge [(A \vee B) \to (C \vee D)]$	$\wedge$I 3, 2

This proof is trivial, but it shows how we can use rules of proof together to demonstrate the validity of an argument form. Also: Using a truth table to show that this argument is valid would have required a staggering 256 lines, since there are eight sentence letters in the argument.

Disjunction

If M were true, then $M \vee N$ would also be true. So the disjunction introduction rule ($\vee$I) allows us to derive a disjunction if we have one of the two disjuncts:

$$\begin{array}{l|ll} m & \mathcal{A} & \\ & \mathcal{A} \vee \mathcal{B} & \vee\text{I } m \\ & \mathcal{B} \vee \mathcal{A} & \vee\text{I } m \end{array}$$

Notice that $\mathcal{B}$ can be *any* sentence whatsoever. So the following is a legitimate proof:

$$\begin{array}{l|ll} 1 & M & \\ \hline 2 & M \vee ([(A \leftrightarrow B) \rightarrow (C \wedge D)] \leftrightarrow [E \wedge F]) & \vee\text{I } 1 \end{array}$$

It may seem odd that just by knowing M we can derive a conclusion that includes sentences like A, B, and the rest— sentences that have nothing to do with M. Yet the conclusion follows immediately by $\vee$I. This is as it should be: The truth conditions for the disjunction mean that, if $\mathcal{A}$ is true, then $\mathcal{A} \vee \mathcal{B}$ is true regardless of what $\mathcal{B}$ is. So the conclusion could not be false if the premise were true; the argument is valid.

Now consider the disjunction elimination rule. What can you conclude from $M \vee N$? You cannot conclude M. It might be M's truth that makes $M \vee N$ true, as in the example above, but it might not. From $M \vee N$ alone, you cannot conclude anything about either M or N specifically. If you also knew that N was false, however, then you would be able to conclude M.

This is just disjunctive syllogism, it will be the disjunction elimination rule ($\vee$E).

$$\begin{array}{l|ll} m & \mathcal{A} \vee \mathcal{B} & \\ n & \neg\mathcal{B} & \\ & \mathcal{A} & \vee\text{E } m, n \end{array} \qquad \begin{array}{l|ll} m & \mathcal{A} \vee \mathcal{B} & \\ n & \neg\mathcal{A} & \\ & \mathcal{B} & \vee\text{E } m, n \end{array}$$

Conditional

Consider this argument:

$$\begin{array}{rl} & R \vee F \\ \therefore & \neg R \rightarrow F \end{array}$$

The argument is certainly a valid one. What should the conditional introduction rule be, such that we can draw this conclusion?

We begin the proof by writing down the premise of the argument and drawing a horizontal line, like this:

$$\begin{array}{l|l}
1 & R \lor F \\ \hline
\end{array}$$

If we had $\neg R$ as a further premise, we could derive F by the $\lor$E rule. We do not have $\neg R$ as a premise of this argument, nor can we derive it directly from the premise we do have— so we cannot simply prove F. What we will do instead is start a *subproof*, a proof within the main proof. When we start a subproof, we draw another vertical line to indicate that we are no longer in the main proof. Then we write in an assumption for the subproof. This can be anything we want. Here, it will be helpful to assume $\neg R$. Our proof now looks like this:

$$\begin{array}{l|l}
1 & R \lor F \\ \hline
2 & \quad \mid \underline{\neg R}
\end{array}$$

It is important to notice that we are not claiming to have proven $\neg R$. We do not need to write in any justification for the assumption line of a subproof. You can think of the subproof as posing the question: What could we show *if* $\neg R$ were true? For one thing, we can derive F. So we do:

$$\begin{array}{l|ll}
1 & R \lor F & \\ \hline
2 & \quad \mid \underline{\neg R} & \\
3 & \quad \mid F & \lor\text{E } 1, 2
\end{array}$$

This has shown that *if* we had $\neg R$ as a premise, *then* we could prove F. In effect, we have proven $\neg R \to F$. So the conditional introduction rule ($\to$I) will allow us to close the subproof and derive $\neg R \to F$ in the main proof. Our final proof looks like this:

$$\begin{array}{l|ll}
1 & R \lor F & \\ \hline
2 & \quad \mid \underline{\neg R} & \\
3 & \quad \mid F & \lor\text{E } 1, 2 \\
4 & \neg R \to F & \to\text{I } 2\text{–}3
\end{array}$$

Notice that the justification for applying the $\to$I rule is the entire subproof. Usually that will be more than just two lines.

It may seem as if the ability to assume anything at all in a subproof would lead to chaos: Does it allow you to prove any conclusion from any premises? The answer is no, it does not. Consider this proof:

1	$\mathcal{A}$	
2	$\quad \mathcal{B}$	
3	$\quad \mathcal{B}$	R 2

It may seem as if this is a proof that you can derive any conclusions $\mathcal{B}$ from any premise $\mathcal{A}$. When the vertical line for the subproof ends, the subproof is *closed*. In order to complete a proof, you must close all of the subproofs. And you cannot close the subproof and use the R rule again on line 4 to derive $\mathcal{B}$ in the main proof. Once you close a subproof, you cannot refer back to individual lines inside it.

Closing a subproof is called *discharging* the assumptions of that subproof. So we can put the point this way: You cannot complete a proof until you have discharged all of the assumptions besides the original premises of the argument.

Of course, it is legitimate to do this:

1	$\mathcal{A}$	
2	$\quad \mathcal{B}$	
3	$\quad \mathcal{B}$	R 2
4	$\mathcal{B} \rightarrow \mathcal{B}$	$\rightarrow$I 2–3

This should not seem so strange, though. Since $\mathcal{B}\rightarrow\mathcal{B}$ is a tautology, no particular premises should be required to validly derive it. (Indeed, as we will see, a tautology follows from any premises.)

Put in a general form, the $\rightarrow$I rule looks like this:

m	$\quad \mathcal{A}$	want $\mathcal{B}$
n	$\quad \mathcal{B}$	
	$\mathcal{A} \rightarrow \mathcal{B}$	$\rightarrow$I m–n

When we introduce a subproof, we typically write what we want to derive in the column. This is just so that we do not forget why we started the subproof if it goes on for five or ten lines. There is no 'want' rule. It is a note to ourselves and not formally part of the proof.

Although it is always permissible to open a subproof with any assumption you please, there is some strategy involved in picking a useful assumption. Starting a subproof with an arbitrary, wacky assumption would just waste lines of the proof. In order to derive a conditional by the →I, for instance, you must assume the antecedent of the conditional in a subproof.

The →I rule also requires that the consequent of the conditional be the last line of the subproof. It is always permissible to close a subproof and discharge its assumptions, but it will not be helpful to do so until you get what you want.

Now consider the conditional elimination rule. Nothing follows from $M \to N$ alone, but if we have both $M \to N$ and M, then we can conclude N. This rule, modus ponens, will be the conditional elimination rule (→E).

m	$\mathcal{A} \to \mathcal{B}$	
n	$\mathcal{A}$	
	$\mathcal{B}$	→E m, n

Now that we have rules for the conditional, consider this argument:

$$\begin{array}{rl} & P \to Q \\ & Q \to R \\ \therefore & P \to R \end{array}$$

We begin the proof by writing the two premises as assumptions. Since the main logical operator in the conclusion is a conditional, we can expect to use the →I rule. For that, we need a subproof— so we write in the antecedent of the conditional as assumption of a subproof:

1	$P \to Q$	
2	$Q \to R$	
3	$\quad P$	

We made P available by assuming it in a subproof, allowing us to use →E on the first premise. This gives us Q, which allows us to use →E on the second premise. Having derived R, we close the subproof. By assuming P we were able to prove R, so we apply the →I rule and finish the proof.

1	$P \to Q$	
2	$Q \to R$	
3	$\quad P$	want R
4	$\quad Q$	→E 1, 3
5	$\quad R$	→E 2, 4
6	$P \to R$	→I 3–5

Biconditional

The rules for the biconditional will be like double-barreled versions of the rules for the conditional.

In order to derive $W \leftrightarrow X$, for instance, you must be able to prove X by assuming W *and* prove W by assuming X. The biconditional introduction rule ($\leftrightarrow$I) requires two subproofs. The subproofs can come in any order, and the second subproof does not need to come immediately after the first— but schematically, the rule works like this:

m	| | $\mathcal{A}$	want $\mathcal{B}$
n	| | $\mathcal{B}$	
p	| | $\mathcal{B}$	want $\mathcal{A}$
q	| | $\mathcal{A}$	
	| $\mathcal{A} \leftrightarrow \mathcal{B}$	$\leftrightarrow$I m–n, p–q

The biconditional elimination rule ($\leftrightarrow$E) lets you do a bit more than the conditional rule. If you have the left-hand subsentence of the biconditional, you can derive the right-hand subsentence. If you have the right-hand subsentence, you can derive the left-hand subsentence. This is the rule:

m	$\mathcal{A} \leftrightarrow \mathcal{B}$		m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{A}$		n	$\mathcal{B}$	
	$\mathcal{B}$	$\leftrightarrow$E m, n		$\mathcal{A}$	$\leftrightarrow$E m, n

Negation

Here is a simple mathematical argument in English:

Assume there is some greatest natural number. Call it A.
That number plus one is also a natural number.
Obviously, $A + 1 > A$.
So there is a natural number greater than A.
This is impossible, since A is assumed to be the greatest natural number.
$\therefore$ There is no greatest natural number.

This argument form is traditionally called a *reductio*. Its full Latin name is *reductio ad absurdum*, which means ‘reduction to absurdity.’ In a reductio, we assume something for the sake of argument— for example, that there is a greatest natural number. Then we show that the assumption leads to two contradictory sentences— for example, that A is the greatest natural number and that it is not. In this way, we show that the original assumption must have been false.

The basic rules for negation will allow for arguments like this. If we assume something and show that it leads to contradictory sentences, then we have proven the negation of the assumption. This is the negation introduction ($\neg$I) rule:

m	$\quad\vert\ \mathcal{A}$	for reductio
n	$\quad\vert\ \mathcal{B}$	
$n+1$	$\quad\vert\ \neg\mathcal{B}$	
$n+2$	$\neg\mathcal{A}$	$\neg$I m–$n+1$

For the rule to apply, the last two lines of the subproof must be an explicit contradiction: some sentence followed on the next line by its negation. We write 'for reductio' as a note to ourselves, a reminder of why we started the subproof. It is not formally part of the proof, and you can leave it out if you find it distracting.

To see how the rule works, suppose we want to prove the law of non-contradiction: $\neg(G \wedge \neg G)$. We can prove this without any premises by immediately starting a subproof. We want to apply $\neg$I to the subproof, so we assume $(G \wedge \neg G)$. We then get an explicit contradiction by $\wedge$E. The proof looks like this:

1	$\quad\vert\ G \wedge \neg G$	for reductio
2	$\quad\vert\ G$	$\wedge$E 1
3	$\quad\vert\ \neg G$	$\wedge$E 1
4	$\neg(G \wedge \neg G)$	$\neg$I 1–3

The $\neg$E rule will work in much the same way. If we assume $\neg\mathcal{A}$ and show that it leads to a contradiction, we have effectively proven $\mathcal{A}$. So the rule looks like this:

m	$\quad\vert\ \neg\mathcal{A}$	for reductio
n	$\quad\vert\ \mathcal{B}$	
$n+1$	$\quad\vert\ \neg\mathcal{B}$	
$n+2$	$\mathcal{A}$	$\neg$E m–$n+1$

4.2 Derived rules

The rules of the natural deduction system are meant to be systematic. There is an introduction and an elimination rule for each logical operator, but why these basic rules rather than some others? Many natural deduction systems have a disjunction elimination rule that works like this:

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\mathcal{A} \to \mathcal{C}$	
o	$\mathcal{B} \to \mathcal{C}$	
	$\mathcal{C}$	DIL m, n, o

Let's call this rule Dilemma (DIL) It might seem as if there will be some proofs that we cannot do with our proof system, because we do not have this as a basic rule. Yet this is not the case. Any proof that you can do using the Dilemma rule can be done with basic rules of our natural deduction system. Consider this proof:

1	$\mathcal{A} \vee \mathcal{B}$	
2	$\mathcal{A} \to \mathcal{C}$	
3	$\mathcal{B} \to \mathcal{C}$	want $\mathcal{C}$
4	$\quad \neg\mathcal{C}$	for reductio
5	$\quad\quad \mathcal{A}$	for reductio
6	$\quad\quad \mathcal{C}$	$\to$E 2, 5
7	$\quad\quad \neg\mathcal{C}$	R 4
8	$\quad \neg\mathcal{A}$	$\neg$I 5–7
9	$\quad\quad \mathcal{B}$	for reductio
10	$\quad\quad \mathcal{C}$	$\to$E 3, 9
11	$\quad\quad \neg\mathcal{C}$	R 4
12	$\quad \mathcal{B}$	$\vee$E 1, 8
13	$\quad \neg\mathcal{B}$	$\neg$I 9–11
14	$\mathcal{C}$	$\neg$E 4–13

$\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are meta-variables. They are not symbols of SL, but stand-ins for arbitrary sentences of SL. So this is not, strictly speaking, a proof in SL. It is more like a recipe. It provides a pattern that can prove anything that the Dilemma rule can prove, using only the basic rules of SL. This means that the

Dilemma rule is not really necessary. Adding it to the list of basic rules would not allow us to derive anything that we could not derive without it.

Nevertheless, the Dilemma rule would be convenient. It would allow us to do in one line what requires eleven lines and several nested subproofs with the basic rules. So we will add it to the proof system as a derived rule.

A DERIVED RULE is a rule of proof that does not make any new proofs possible. Anything that can be proven with a derived rule can be proven without it. You can think of a short proof using a derived rule as shorthand for a longer proof that uses only the basic rules. Anytime you use the Dilemma rule, you could always take ten extra lines and prove the same thing without it.

For the sake of convenience, we will add several other derived rules. One is *modus tollens* (MT).

m	$\mathcal{A} \to \mathcal{B}$	
n	$\neg\mathcal{B}$	
	$\neg\mathcal{A}$	MT m, n

We leave the proof of this rule as an exercise. Note that if we had already proven the MT rule, then the proof of the DIL rule could have been done in only five lines.

We also add hypothetical syllogism (HS) as a derived rule. We have already given a proof of it on p. 53.

m	$\mathcal{A} \to \mathcal{B}$	
n	$\mathcal{B} \to \mathcal{C}$	
	$\mathcal{A} \to \mathcal{C}$	HS m, n

4.3 Rules of replacement

Consider how you would prove this argument: $F \to (G \wedge H)$, $\therefore F \to G$

Perhaps it is tempting to write down the premise and apply the $\wedge$E rule to the conjunction $(G \wedge H)$. This is impermissible, however, because the basic rules of proof can only be applied to whole sentences. We need to get $(G \wedge H)$ on a line by itself. We can prove the argument in this way:

1	$F \to (G \wedge H)$	
2	$\quad F$	want G
3	$\quad G \wedge H$	$\to$E 1, 2
4	$\quad G$	$\wedge$E 3
5	$F \to G$	$\to$I 2–4

We will now introduce some derived rules that may be applied to part of a sentence. These are called RULES OF REPLACEMENT, because they can be used to replace part of a sentence with a logically equivalent expression. One simple rule of replacement is commutivity (abbreviated Comm), which says that we can swap the order of conjuncts in a conjunction or the order of disjuncts in a disjunction. We define the rule this way:

$$\begin{array}{rl} (\mathcal{A} \wedge \mathcal{B}) \Longleftrightarrow (\mathcal{B} \wedge \mathcal{A}) & \\ (\mathcal{A} \vee \mathcal{B}) \Longleftrightarrow (\mathcal{B} \vee \mathcal{A}) & \\ (\mathcal{A} \leftrightarrow \mathcal{B}) \Longleftrightarrow (\mathcal{B} \leftrightarrow \mathcal{A}) & \quad \text{Comm} \end{array}$$

The bold arrow means that you can take a subformula on one side of the arrow and replace it with the subformula on the other side. The arrow is double-headed because rules of replacement work in both directions.

Consider this argument: $(M \vee P) \to (P \wedge M)$, $\therefore$ $(P \vee M) \to (M \wedge P)$

It is possible to give a proof of this using only the basic rules, but it will be long and inconvenient. With the Comm rule, we can provide a proof easily:

1	$(M \vee P) \to (P \wedge M)$	
2	$(P \vee M) \to (P \wedge M)$	Comm 1
3	$(P \vee M) \to (M \wedge P)$	Comm 2

Another rule of replacement is double negation (DN). With the DN rule, you can remove or insert a pair of negations anywhere in a sentence. This is the rule:

$$\neg\neg\mathcal{A} \Longleftrightarrow \mathcal{A} \quad \text{DN}$$

Two more replacement rules are called De Morgan's Laws, named for the 19th-century British logician August De Morgan. (Although De Morgan did discover these laws, he was not the first to do so.) The rules capture useful relations between negation, conjunction, and disjunction. Here are the rules, which we abbreviate DeM:

$$\neg(\mathcal{A} \vee \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \wedge \neg\mathcal{B})$$
$$\neg(\mathcal{A} \wedge \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \vee \neg\mathcal{B}) \quad \text{DeM}$$

Because $\mathcal{A} \to \mathcal{B}$ is a *material conditional*, it is equivalent to $\neg\mathcal{A} \vee \mathcal{B}$. A further replacement rule captures this equivalence. We abbreviate the rule MC, for 'material conditional.' It takes two forms:

$$(\mathcal{A} \to \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \vee \mathcal{B})$$
$$(\mathcal{A} \vee \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \to \mathcal{B}) \quad \text{MC}$$

Now consider this argument: $\neg(P \to Q)$, $\therefore P \wedge \neg Q$

As always, we could prove this argument using only the basic rules. With rules of replacement, though, the proof is much simpler:

1	$\neg(P \to Q)$	
2	$\neg(\neg P \vee Q)$	MC 1
3	$\neg\neg P \wedge \neg Q$	DeM 2
4	$P \wedge \neg Q$	DN 3

A final replacement rule captures the relation between conditionals and biconditionals. We will call this rule biconditional exchange and abbreviate it $\leftrightarrow$ex.

$$[(\mathcal{A} \to \mathcal{B}) \wedge (\mathcal{B} \to \mathcal{A})] \Longleftrightarrow (\mathcal{A} \leftrightarrow \mathcal{B}) \quad \leftrightarrow\text{ex}$$

4.4 Proof-theoretic concepts

We will use the symbol '$\vdash$' to indicate that a proof is possible. This symbol is called the *turnstile*. Sometimes it is called a *single turnstile*, to underscore the fact that this is not the double turnstile symbol ($\models$) that we used to represent semantic entailment in ch. 6.

When we write $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash \mathcal{B}$, this means that it is possible to give a proof of $\mathcal{B}$ with $\mathcal{A}_1$,$\mathcal{A}_2$,... as premises. With just one premise, we leave out the curly braces, so $\mathcal{A} \vdash \mathcal{B}$ means that there is a proof of $\mathcal{B}$ with $\mathcal{A}$ as a premise. Naturally, $\vdash \mathcal{C}$ means that there is a proof of $\mathcal{C}$ that has no premises.

Often, logical proofs are called *derivations*. So $\mathcal{A} \vdash \mathcal{B}$ can be read as '$\mathcal{B}$ is derivable from $\mathcal{A}$.'

A THEOREM is a sentence that is derivable without any premises; i.e., $\mathcal{T}$ is a theorem if and only if $\vdash \mathcal{T}$.

It is not too hard to show that something is a theorem— you just have to give a proof of it. How could you show that something is *not* a theorem? If its negation is a theorem, then you could provide a proof. For example, it is easy to prove $\neg(Pa \wedge \neg Pa)$, which shows that $(Pa \wedge \neg Pa)$ cannot be a theorem. For a sentence that is neither a theorem nor the negation of a theorem, however, there is no easy way to show this. You would have to demonstrate not just that certain proof strategies fail, but that no proof is possible. Even if you fail in trying to prove a sentence in a thousand different ways, perhaps the proof is just too long and complex for you to make out.

Two sentences $\mathcal{A}$ and $\mathcal{B}$ are PROVABLY EQUIVALENT if and only if each can be derived from the other; i.e., $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$

It is relatively easy to show that two sentences are provably equivalent— it just requires a pair of proofs. Showing that sentences are *not* provably equivalent would be much harder. It would be just as hard as showing that a sentence is not a theorem. (In fact, these problems are interchangeable. Can you think of a sentence that would be a theorem if and only if $\mathcal{A}$ and $\mathcal{B}$ were provably equivalent?)

The set of sentences $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\}$ is PROVABLY INCONSISTENT if and only if a contradiction is derivable from it; i.e., for some sentence $\mathcal{B}$, $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash \mathcal{B}$ and $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash \neg\mathcal{B}$.

It is easy to show that a set is provably inconsistent: You just need to assume the sentences in the set and prove a contradiction. Showing that a set is *not* provably inconsistent will be much harder. It would require more than just providing a proof or two; it would require showing that proofs of a certain kind are *impossible*.

Practice Exercises

⋆ **Part A** Provide a justification (rule and line numbers) for each line of proof that requires one.

$$
\begin{array}{r|l}
1 & W \rightarrow \neg B \\
2 & A \wedge W \\
3 & \underline{B \vee (J \wedge K)} \\
4 & W \\
5 & \neg B \\
6 & J \wedge K \\
7 & K
\end{array}
\qquad\qquad
\begin{array}{r|l}
1 & L \leftrightarrow \neg O \\
2 & \underline{L \vee \neg O} \\
3 & \quad \begin{array}{|l} \underline{\neg L} \end{array} \\
4 & \quad \begin{array}{|l} \neg O \end{array} \\
5 & \quad \begin{array}{|l} L \end{array} \\
6 & \quad \begin{array}{|l} \neg L \end{array} \\
7 & L
\end{array}
$$

$$
\begin{array}{r|l}
1 & Z \to (C \wedge \neg N) \\
2 & \neg Z \to (N \wedge \neg C) \\ \hline
3 & \quad \mid \neg(N \vee C) \\
4 & \quad \mid \neg N \wedge \neg C \\
5 & \quad \mid \quad \mid Z \\
6 & \quad \mid \quad \mid C \wedge \neg N \\
7 & \quad \mid \quad \mid C \\
8 & \quad \mid \quad \mid \neg C \\
9 & \quad \mid \neg Z \\
10 & \quad \mid N \wedge \neg C \\
11 & \quad \mid N \\
12 & \quad \mid \neg N \\
13 & N \vee C
\end{array}
$$

⋆ **Part B** Give a proof for each argument in SL.

1. $K \wedge L, \therefore K \leftrightarrow L$
2. $A \to (B \to C), \therefore (A \wedge B) \to C$
3. $P \wedge (Q \vee R), P \to \neg R, \therefore Q \vee E$
4. $(C \wedge D) \vee E, \therefore E \vee D$
5. $\neg F \to G, F \to H, \therefore G \vee H$
6. $(X \wedge Y) \vee (X \wedge Z), \neg(X \wedge D), D \vee M \therefore M$

Part C Give a proof for each argument in SL.

1. $Q \to (Q \wedge \neg Q), \therefore \neg Q$
2. $J \to \neg J, \therefore \neg J$
3. $E \vee F, F \vee G, \neg F, \therefore E \wedge G$
4. $A \leftrightarrow B, B \leftrightarrow C, \therefore A \leftrightarrow C$
5. $M \vee (N \to M), \therefore \neg M \to \neg N$
6. $S \leftrightarrow T, \therefore S \leftrightarrow (T \vee S)$
7. $(M \vee N) \wedge (O \vee P), N \to P, \neg P, \therefore M \wedge O$
8. $(Z \wedge K) \vee (K \wedge M), K \to D, \therefore D$

Part D Show that each of the following sentences is a theorem in SL.

1. $O \to O$

2. $N \vee \neg N$
3. $\neg(P \wedge \neg P)$
4. $\neg(A \to \neg C) \to (A \to C)$
5. $J \leftrightarrow [J \vee (L \wedge \neg L)]$

Part E Show that each of the following pairs of sentences are provably equivalent in SL.

1. $\neg\neg\neg\neg G$, G
2. $T \to S$, $\neg S \to \neg T$
3. $R \leftrightarrow E$, $E \leftrightarrow R$
4. $\neg G \leftrightarrow H$, $\neg(G \leftrightarrow H)$
5. $U \to I$, $\neg(U \wedge \neg I)$

Part F Provide proofs to show each of the following.

1. $M \wedge (\neg N \to \neg M) \vdash (N \wedge M) \vee \neg M$
2. $\{C \to (E \wedge G), \neg C \to G\} \vdash G$
3. $\{(Z \wedge K) \leftrightarrow (Y \wedge M), D \wedge (D \to M)\} \vdash Y \to Z$
4. $\{(W \vee X) \vee (Y \vee Z), X \to Y, \neg Z\} \vdash W \vee Y$

Part G For the following, provide proofs using only the basic rules. The proofs will be longer than proofs of the same claims would be using the derived rules.

1. Show that MT is a legitimate derived rule. Using only the basic rules, prove the following: $\mathcal{A} \to \mathcal{B}$, $\neg\mathcal{B}$, $\therefore \neg\mathcal{A}$
2. Show that Comm is a legitimate rule for the biconditional. Using only the basic rules, prove that $\mathcal{A} \leftrightarrow \mathcal{B}$ and $\mathcal{B} \leftrightarrow \mathcal{A}$ are equivalent.
3. Using only the basic rules, prove the following instance of DeMorgan's Laws: $(\neg A \wedge \neg B)$, $\therefore \neg(A \vee B)$
4. Show that $\leftrightarrow$ex is a legitimate derived rule. Using only the basic rules, prove that $D \leftrightarrow E$ and $(D \to E) \wedge (E \to D)$ are equivalent.

Chapter 5

Quantified logic

This chapter introduces a logical language called QL. It is a version of *quantified logic*, because it allows for quantifiers like *all* and *some*. Quantified logic is also sometimes called *predicate logic*, because the basic units of the language are predicates and terms.

5.1 From sentences to predicates

Consider the following argument, which is obviously valid in English:

> If everyone knows logic, then either no one will be confused or everyone will. Everyone will be confused only if we try to believe a contradiction. This is a logic class, so everyone knows logic.
> ∴ If we don't try to believe a contradiction, then no one will be confused.

In order to symbolize this in SL, we will need a symbolization key.

L: Everyone knows logic.
N: No one will be confused.
E: Everyone will be confused.
B: We try to believe a contradiction.

Notice that N and E are both about people being confused, but they are two separate sentence letters. We could not replace E with $\neg N$. Why not? $\neg N$ means 'It is not the case that no one will be confused.' This would be the case if even one person were confused, so it is a long way from saying that *everyone* will be confused.

Once we have separate sentence letters for N and E, however, we erase any connection between the two. They are just two atomic sentences which might be true or false independently. In English, it could never be the case that both no one and everyone was confused. As sentences of SL, however, there is a truth-value assignment for which N and E are both true.

Expressions like 'no one', 'everyone', and 'anyone' are called *quantifiers*. By translating N and E as separate atomic sentences, we leave out the *quantifier structure* of the sentences. Fortunately, the quantifier structure is not what makes this argument valid. As such, we can safely ignore it. To see this, we translate the argument to SL:

$$\begin{array}{rl} & L \rightarrow (N \vee E) \\ & E \rightarrow B \\ & L \\ \therefore & \neg B \rightarrow N \end{array}$$

This is a valid argument in SL. (You can do a truth table to check this.)

Now consider another argument. This one is also valid in English.

Willard is a logician. All logicians wear funny hats.
∴ Willard wears a funny hat.

To symbolize it in SL, we define a symbolization key:

L: Willard is a logician.
A: All logicians wear funny hats.
F: Willard wears a funny hat.

Now we symbolize the argument:

$$\begin{array}{rl} & L \\ & A \\ \therefore & F \end{array}$$

This is *invalid* in SL. (Again, you can confirm this with a truth table.) There is something very wrong here, because this is clearly a valid argument in English. The symbolization in SL leaves out all the important structure. Once again, the translation to SL overlooks quantifier structure: The sentence 'All logicians wear funny hats' is about both logicians and hat-wearing. By not translating this structure, we lose the connection between Willard's being a logician and Willard's wearing a hat.

Some arguments with quantifier structure can be captured in SL, like the first example, even though SL ignores the quantifier structure. Other arguments are completely botched in SL, like the second example. Notice that the problem is not that we have made a mistake while symbolizing the second argument. These are the best symbolizations we can give for these arguments *in SL*.

Generally, if an argument containing quantifiers comes out *valid in SL*, then the English language argument is valid. If it comes out *invalid in SL*, then we cannot say the English language argument is invalid. The argument might be valid because of quantifier structure which the natural language argument has and which the argument in SL lacks.

Similarly, if a sentence with quantifiers comes out as a *tautology in SL*, then the English sentence is logically true. If it comes out as *contingent in SL*, then this might be because of the structure of the quantifiers that gets removed when we translate into the formal language.

In order to symbolize arguments that rely on quantifier structure, we need to develop a different logical language. We will call this language quantified logic, QL.

5.2 Building blocks of QL

Just as sentences were the basic unit of sentential logic, predicates will be the basic unit of quantified logic. A predicate is an expression like 'is a dog.' This is not a sentence on its own. It is neither true nor false. In order to be true or false, we need to specify something: Who or what is it that is a dog?

The details of this will be explained in the rest of the chapter, but here is the basic idea: In QL, we will represent predicates with capital letters. For instance, we might let D stand for '_____ is a dog.' We will use lower-case letters as the names of specific things. For instance, we might let b stand for Bertie. The expression Db will be a sentence in QL. It is a translation of the sentence 'Bertie is a dog.'

In order to represent quantifier structure, we will also have symbols that represent quantifiers. For instance, '$\exists$' will mean 'There is some_____.' So to say that there is a dog, we can write $\exists x Dx$; that is: There is some x such that x is a dog.

That will come later. We start by defining singular terms and predicates.

Singular Terms

In English, a SINGULAR TERM is a word or phrase that refers to a *specific* person, place, or thing. The word 'dog' is not a singular term, because there are a great many dogs. The phrase 'Philip's dog Bertie' is a singular term, because it refers to a specific little terrier.

A PROPER NAME is a singular term that picks out an individual without describing it. The name 'Emerson' is a proper name, and the name alone does not tell you anything about Emerson. Of course, some names are traditionally given to boys and other are traditionally given to girls. If 'Jack Hathaway' is used as a singular term, you might guess that it refers to a man. However, the name does not necessarily mean that the person referred to is a man— or even that the creature referred to is a person. Jack might be a giraffe for all you could tell just from the name. There is a great deal of philosophical action surrounding this issue, but the important point here is that a name is a singular term because it picks out a single, specific individual.

Other singular terms more obviously convey information about the thing to which they refer. For instance, you can tell without being told anything further that 'Philip's dog Bertie' is a singular term that refers to a dog. A DEFINITE DESCRIPTION picks out an individual by means of a unique description. In English, definite descriptions are often phrases of the form 'the such-and-so.' They refer to *the* specific thing that matches the given description. For example, 'the tallest member of Monty Python' and 'the first emperor of China' are definite descriptions. A description that does not pick out a specific individual is not a definite description. 'A member of Monty Python' and 'an emperor of China' are not definite descriptions.

We can use proper names and definite descriptions to pick out the same thing. The proper name 'Mount Rainier' names the location picked out by the definite description 'the highest peak in Washington state.' The expressions refer to the same place in different ways. You learn nothing from my saying that I am going to Mount Rainier, unless you already know some geography. You could guess that it is a mountain, perhaps, but even this is not a sure thing; for all you know it might be a college, like Mount Holyoke. Yet if I were to say that I was going to the highest peak in Washington state, you would know immediately that I was going to a mountain in Washington state.

In English, the specification of a singular term may depend on context; 'Willard' means a specific person and not just someone named Willard; 'P.D. Magnus' as a logical singular term means *me* and not the other P.D. Magnus. We live with this kind of ambiguity in English, but it is important to keep in mind that singular terms in QL must refer to just one specific thing.

In QL, we will symbolize singular terms with lower-case letters a through w. We can add subscripts if we want to use some letter more than once. So $a, b, c, \ldots w, a_1, f_{32}, j_{390}$, and m_{12} are all terms in QL.

Singular terms are called CONSTANTS because they pick out specific individuals. Note that x, y, and z are not constants in QL. They will be VARIABLES, letters which do not stand for any specific thing. We will need them when we introduce quantifiers.

Predicates

The simplest predicates are properties of individuals. They are things you can say about an object. '_____ is a dog' and '_____ is a member of Monty Python' are both predicates. In translating English sentences, the term will not always come at the beginning of the sentence: 'A piano fell on _____' is also a predicate. Predicates like these are called ONE-PLACE or MONADIC, because there is only one blank to fill in. A one-place predicate and a singular term combine to make a sentence.

Other predicates are about the *relation* between two things. For instance, '_____ is bigger than _____', '_____ is to the left of _____', and '_____ owes money to _____.' These are TWO-PLACE or DYADIC predicates, because they need to be filled in with two terms in order to make a sentence.

In general, you can think about predicates as schematic sentences that need to be filled out with some number of terms. Conversely, you can start with sentences and make predicates out of them by removing terms. Consider the sentence, 'Vinnie borrowed the family car from Nunzio.' By removing a singular term, we can recognize this sentence as using any of three different monadic predicates:

_____borrowed the family car from Nunzio.
Vinnie borrowed _____from Nunzio.
Vinnie borrowed the family car from _____.

By removing two singular terms, we can recognize three different dyadic predicates:

Vinnie borrowed _____ from _____.
_____ borrowed the family car from _____.
_____ borrowed _____ from Nunzio.

By removing all three singular terms, we can recognize one THREE-PLACE or TRIADIC predicate:

_____ borrowed _____ from _____.

If we are translating this sentence into QL, should we translate it with a one-, two-, or three-place predicate? It depends on what we want to be able to say. If the only thing that we will discuss being borrowed is the family car, then the generality of the three-place predicate is unnecessary. If the only borrowing we need to symbolize is different people borrowing the family car from Nunzio, then a one-place predicate will be enough.

In general, we can have predicates with as many places as we need. Predicates with more than one place are called POLYADIC. Predicates with n places, for some number n, are called N-PLACE or N-ADIC.

In QL, we symbolize predicates with capital letters A through Z, with or without subscripts. When we give a symbolization key for predicates, we will not use blanks; instead, we will use variables. By convention, constants are listed at the end of the key. So we might write a key that looks like this:

Ax: x is angry.
Hx: x is happy.
$\mathbf{T_1}$xy: x is as tall or taller than y.
$\mathbf{T_2}$xy: x is as tough or tougher than y.
Bxyz: y is between x and z.
d: Donald
g: Gregor
m: Marybeth

We can symbolize sentences that use any combination of these predicates and terms. For example:

1. Donald is angry.
2. If Donald is angry, then so are Gregor and Marybeth.
3. Marybeth is at least as tall and as tough as Gregor.
4. Donald is shorter than Gregor.
5. Gregor is between Donald and Marybeth.

Sentence 1 is straightforward: Ad. The 'x' in the key entry 'Ax' is just a placeholder; we can replace it with other terms when translating.

Sentence 2 can be paraphrased as, 'If Ad, then Ag and Am.' QL has all the truth-functional connectives of SL, so we translate this as $Ad \rightarrow (Ag \wedge Am)$.

Sentence 3 can be translated as $T_1mg \wedge T_2mg$.

Sentence 4 might seem as if it requires a new predicate. If we only needed to symbolize this sentence, we could define a predicate like Sxy to mean 'x is shorter than y.' However, this would ignore the logical connection between 'shorter' and 'taller.' Considered only as symbols of QL, there is no connection between S and T_1. They might mean anything at all. Instead of introducing a new predicate, we paraphrase sentence 4 using predicates already in our key: 'It is not the case that Donald is as tall or taller than Gregor.' We can translate it as $\neg T_1dg$.

Sentence 5 requires that we pay careful attention to the order of terms in the key. It becomes $Bdgm$.

5.3 Quantifiers

We are now ready to introduce quantifiers. Consider these sentences:

6. Everyone is happy.
7. Everyone is at least as tough as Donald.
8. Someone is angry.

It might be tempting to translate sentence 6 as $Hd \wedge Hg \wedge Hm$. Yet this would only say that Donald, Gregor, and Marybeth are happy. We want to say that *everyone* is happy, even if we have not defined a constant to name them. In order to do this, we introduce the '$\forall$' symbol. This is called the UNIVERSAL QUANTIFIER.

A quantifier must always be followed by a variable and a formula that includes that variable. We can translate sentence 6 as $\forall x Hx$. Paraphrased in English, this means 'For all x, x is happy.' We call $\forall x$ an *x-quantifier*. The formula that follows the quantifier is called the *scope* of the quantifier. We will give a formal definition of scope later, but intuitively it is the part of the sentence that the quantifier quantifies over. In $\forall x Hx$, the scope of the universal quantifier is Hx.

Sentence 7 can be paraphrased as, 'For all x, x is at least as tough as Donald.' This translates as $\forall x T_2 xd$.

In these quantified sentences, the variable x is serving as a kind of placeholder. The expression $\forall x$ means that you can pick anyone and put them in as x. There is no special reason to use x rather than some other variable. The sentence $\forall x Hx$ means exactly the same thing as $\forall y Hy$, $\forall z Hz$, and $\forall x_5 Hx_5$.

To translate sentence 8, we introduce another new symbol: the EXISTENTIAL QUANTIFIER, $\exists$. Like the universal quantifier, the existential quantifier requires a variable. Sentence 8 can be translated as $\exists x Ax$. This means that there is some x which is angry. More precisely, it means that there is *at least one* angry person. Once again, the variable is a kind of placeholder; we could just as easily have translated sentence 8 as $\exists z Az$.

Consider these further sentences:

9. No one is angry.
10. There is someone who is not happy.
11. Not everyone is happy.

Sentence 9 can be paraphrased as, 'It is not the case that someone is angry.' This can be translated using negation and an existential quantifier: $\neg \exists x Ax$. Yet sentence 9 could also be paraphrased as, 'Everyone is not angry.' With this in mind, it can be translated using negation and a universal quantifier: $\forall x \neg Ax$.

Both of these are acceptable translations, because they are logically equivalent. The critical thing is whether the negation comes before or after the quantifier.

In general, $\forall x\mathcal{A}$ is logically equivalent to $\neg\exists x\neg\mathcal{A}$. This means that any sentence which can be symbolized with a universal quantifier can be symbolized with an existential quantifier, and vice versa. One translation might seem more natural than the other, but there is no logical difference in translating with one quantifier rather than the other. For some sentences, it will simply be a matter of taste.

Sentence 10 is most naturally paraphrased as, 'There is some x such that x is not happy.' This becomes $\exists x\neg Hx$. Equivalently, we could write $\neg\forall xHx$.

Sentence 11 is most naturally translated as $\neg\forall xHx$. This is logically equivalent to sentence 10 and so could also be translated as $\exists x\neg Hx$.

Although we have two quantifiers in QL, we could have an equivalent formal language with only one quantifier. We could proceed with only the universal quantifier, for instance, and treat the existential quantifier as a notational convention. We use square brackets [ ] to make some sentences more readable, but we know that these are really just parentheses ( ). In the same way, we could write '$\exists x$' knowing that this is just shorthand for '$\neg\forall x\neg$.' There is a choice between making logic formally simple and making it expressively simple. With QL, we opt for expressive simplicity. Both $\forall$ and $\exists$ will be symbols of QL.

Universe of Discourse

Given the symbolization key we have been using, $\forall xHx$ means 'Everyone is happy.' Who is included in this *everyone*? When we use sentences like this in English, we usually do not mean everyone now alive on the Earth. We certainly do not mean everyone who was ever alive or who will ever live. We mean something more modest: everyone in the building, everyone in the class, or everyone in the room.

In order to eliminate this ambiguity, we will need to specify a UNIVERSE OF DISCOURSE— abbreviated UD. The UD is the set of things that we are talking about. So if we want to talk about people in Chicago, we define the UD to be people in Chicago. We write this at the beginning of the symbolization key, like this:

UD: people in Chicago

The quantifiers *range over* the universe of discourse. Given this UD, $\forall x$ means 'Everyone in Chicago' and $\exists x$ means 'Someone in Chicago.' Each constant names some member of the UD, so we can only use this UD with the symbolization key above if Donald, Gregor, and Marybeth are all in Chicago. If we want to talk about people in places besides Chicago, then we need to include those people in the UD.

In QL, the UD must be *non-empty*; that is, it must include at least one thing. It is possible to construct formal languages that allow for empty UDs, but this introduces complications.

Even allowing for a UD with just one member can produce some strange results. Suppose we have this as a symbolization key:

UD: the Eiffel Tower
Px: x is in Paris.

The sentence $\forall xPx$ might be paraphrased in English as 'Everything is in Paris.' Yet that would be misleading. It means that everything *in the UD* is in Paris. This UD contains only the Eiffel Tower, so with this symbolization key $\forall xPx$ just means that the Eiffel Tower is in Paris.

Non-referring terms

In QL, each constant must pick out exactly one member of the UD. A constant cannot refer to more than one thing— it is a *singular* term. Each constant must still pick out *something.* This is connected to a classic philosophical problem: the so-called problem of non-referring terms.

Medieval philosophers typically used sentences about the *chimera* to exemplify this problem. Chimera is a mythological creature; it does not really exist. Consider these two sentences:

12. Chimera is angry.
13. Chimera is not angry.

It is tempting just to define a constant to mean 'chimera.' The symbolization key would look like this:

UD: creatures on Earth
Ax: x is angry.
c: chimera

We could then translate sentence 12 as Ac and sentence 13 as $\neg Ac$.

Problems will arise when we ask whether these sentences are true or false.

One option is to say that sentence 12 is not true, because there is no chimera. If sentence 12 is false because it talks about a non-existent thing, then sentence

13 is false for the same reason. Yet this would mean that Ac and $\neg Ac$ would both be false. Given the truth conditions for negation, this cannot be the case.

Since we cannot say that they are both false, what should we do? Another option is to say that sentence 12 is *meaningless* because it talks about a non-existent thing. So Ac would be a meaningful expression in QL for some interpretations but not for others. Yet this would make our formal language hostage to particular interpretations. Since we are interested in logical form, we want to consider the logical force of a sentence like Ac apart from any particular interpretation. If Ac were sometimes meaningful and sometimes meaningless, we could not do that.

This is the *problem of non-referring terms*, and we will return to it later (see p. 88.) The important point for now is that each constant of QL *must* refer to something in the UD, although the UD can be any set of things that we like. If we want to symbolize arguments about mythological creatures, then we must define a UD that includes them. This option is important if we want to consider the logic of stories. We can translate a sentence like 'Sherlock Holmes lived at 221B Baker Street' by including fictional characters like Sherlock Holmes in our UD.

5.4 Translating to QL

We now have all of the pieces of QL. Translating more complicated sentences will only be a matter of knowing the right way to combine predicates, constants, quantifiers, variables, and connectives. Consider these sentences:

14. Every coin in my pocket is a quarter.
15. Some coin on the table is a dime.
16. Not all the coins on the table are dimes.
17. None of the coins in my pocket are dimes.

In providing a symbolization key, we need to specify a UD. Since we are talking about coins in my pocket and on the table, the UD must at least contain all of those coins. Since we are not talking about anything besides coins, we let the UD be all coins. Since we are not talking about any specific coins, we do not need to define any constants. So we define this key:

UD: all coins
Px: x is in my pocket.
Tx: x is on the table.
Qx: x is a quarter.
Dx: x is a dime.

Sentence 14 is most naturally translated with a universal quantifier. The universal quantifier says something about everything in the UD, not just about the coins in my pocket. Sentence 14 means that (for any coin) *if* that coin is in my pocket, *then* it is a quarter. So we can translate it as $\forall x(Px \rightarrow Qx)$.

Since sentence 14 is about coins that are both in my pocket *and* that are quarters, it might be tempting to translate it using a conjunction. However, the sentence $\forall x(Px \wedge Qx)$ would mean that everything in the UD is both in my pocket and a quarter: All the coins that exist are quarters in my pocket. This would be a crazy thing to say, and it means something very different than sentence 14.

Sentence 15 is most naturally translated with an existential quantifier. It says that there is some coin which is both on the table and which is a dime. So we can translate it as $\exists x(Tx \wedge Dx)$.

Notice that we needed to use a conditional with the universal quantifier, but we used a conjunction with the existential quantifier. What would it mean to write $\exists x(Tx \rightarrow Dx)$? Probably not what you think. It means that there is some member of the UD which would satisfy the subformula; roughly speaking, there is some a such that $(Ta \rightarrow Da)$ is true. In SL, $\mathcal{A} \rightarrow \mathcal{B}$ is logically equivalent to $\neg\mathcal{A} \vee \mathcal{B}$, and this will also hold in QL. So $\exists x(Tx \rightarrow Dx)$ is true if there is some a such that $(\neg Ta \vee Da)$; i.e., it is true if some coin is *either* not on the table *or* is a dime. Of course there is a coin that is not on the table— there are coins in lots of other places. So $\exists x(Tx \rightarrow Dx)$ is trivially true. A conditional will usually be the natural connective to use with a universal quantifier, but a conditional within the scope of an existential quantifier can do very strange things. As a general rule, do not put conditionals in the scope of existential quantifiers unless you are sure that you need one.

Sentence 16 can be paraphrased as, 'It is not the case that every coin on the table is a dime.' So we can translate it as $\neg\forall x(Tx \rightarrow Dx)$. You might look at sentence 16 and paraphrase it instead as, 'Some coin on the table is not a dime.' You would then translate it as $\exists x(Tx \wedge \neg Dx)$. Although it is probably not obvious, these two translations are logically equivalent. (This is due to the logical equivalence between $\neg\forall x\mathcal{A}$ and $\exists x\neg\mathcal{A}$, along with the equivalence between $\neg(\mathcal{A} \rightarrow \mathcal{B})$ and $\mathcal{A} \wedge \neg\mathcal{B}$.)

Sentence 17 can be paraphrased as, 'It is not the case that there is some dime in my pocket.' This can be translated as $\neg\exists x(Px \wedge Dx)$. It might also be paraphrased as, 'Everything in my pocket is a non-dime,' and then could be translated as $\forall x(Px \rightarrow \neg Dx)$. Again the two translations are logically equivalent. Both are correct translations of sentence 17.

We can now translate the argument from p. 64, the one that motivated the need for quantifiers:

Willard is a logician. All logicians wear funny hats.
$\therefore$ Willard wears a funny hat.

UD: people
Lx: x is a logician.
Fx: x wears a funny hat.
w: Willard

Translating, we get:

$$\begin{array}{ll} & Lw \\ & \forall x(Lx \to Fx) \\ \therefore & Fw \end{array}$$

This captures the structure that was left out of the SL translation of this argument, and this is a valid argument in QL.

Empty predicates

A predicate need not apply to anything in the UD. A predicate that applies to nothing in the UD is called an EMPTY predicate.

Suppose we want to symbolize these two sentences:

18. Every monkey knows sign language.
19. Some monkey knows sign language.

It is possible to write the symbolization key for these sentences in this way:

UD: animals
Mx: x is a monkey.
Sx: x knows sign language.

Sentence 18 can now be translated as $\forall x(Mx \to Sx)$.

Sentence 19 becomes $\exists x(Mx \wedge Sx)$.

It is tempting to say that sentence 18 entails sentence 19; that is: if every monkey knows sign language, then it must be that some monkey knows sign language. This is a valid inference in Aristotelean logic: All Ms are S, $\therefore$ some M is S. However, the entailment does not hold in QL. It is possible for the sentence $\forall x(Mx \to Sx)$ to be true even though the sentence $\exists x(Mx \wedge Sx)$ is false.

How can this be? The answer comes from considering whether these sentences would be true or false *if there were no monkeys.*

We have defined $\forall$ and $\exists$ in such a way that $\forall\mathcal{A}$ is equivalent to $\neg\exists\neg\mathcal{A}$. As such, the universal quantifier doesn't involve the existence of anything— only non-existence. If sentence 18 is true, then there are *no* monkeys who don't know sign language. If there were no monkeys, then $\forall x(Mx \rightarrow Sx)$ would be true and $\exists x(Mx \wedge Sx)$ would be false.

We allow empty predicates because we want to be able to say things like, 'I do not know if there are any monkeys, but any monkeys that there are know sign language.' That is, we want to be able to have predicates that do not (or might not) refer to anything.

What happens if we add an empty predicate R to the interpretation above? For example, we might define Rx to mean 'x is a refrigerator.' Now the sentence $\forall x(Rx \rightarrow Mx)$ will be true. This is counterintuitive, since we do not want to say that there are a whole bunch of refrigerator monkeys. It is important to remember, though, that $\forall x(Rx \rightarrow Mx)$ means that any member of the UD which is a refrigerator is a monkey. Since the UD is animals, there are no refrigerators in the UD and so the sentence is trivially true.

If you were actually translating the sentence 'All refrigerators are monkeys', then you would want to include appliances in the UD. Then the predicate R would not be empty and the sentence $\forall x(Rx \rightarrow Mx)$ would be false.

- ▷ A UD must have *at least* one member.
- ▷ A predicate may apply to some, all, or no members of the UD.
- ▷ A constant must pick out *exactly* one member of the UD.
  A member of the UD may be picked out by one constant, many constants, or none at all.

Picking a Universe of Discourse

The appropriate symbolization of an English language sentence in QL will depend on the symbolization key. In some ways, this is obvious: It matters whether Dx means 'x is dainty' or 'x is dangerous.' The meaning of sentences in QL also depends on the UD.

Let Rx mean 'x is a rose,' let Tx mean 'x has a thorn,' and consider this sentence:

20. Every rose has a thorn.

It is tempting to say that sentence 20 should be translated as $\forall x(Rx \rightarrow Tx)$. If the UD contains all roses, that would be correct. Yet if the UD is merely *things* *on my kitchen table*, then $\forall x(Rx \rightarrow Tx)$ would only mean that every rose on my kitchen table has a thorn. If there are no roses on my kitchen table, the sentence would be trivially true.

The universal quantifier only ranges over members of the UD, so we need to include all roses in the UD in order to translate sentence 20. We have two options. First, we can restrict the UD to include all roses but *only* roses. Then sentence 20 becomes $\forall xTx$. This means that everything in the UD has a thorn; since the UD just is the set of roses, this means that every rose has a thorn. This option can save us trouble if every sentence that we want to translate using the symbolization key is about roses.

Second, we can let the UD contain things besides roses: rhododendrons, rats, rifles, and whatall else. Then sentence 20 must be $\forall x(Rx \rightarrow Tx)$.

If we wanted the universal quantifier to mean *every* thing, without restriction, then we might try to specify a UD that contains everything. This would lead to problems. Does 'everything' include things that have only been imagined, like fictional characters? On the one hand, we want to be able to symbolize arguments about Hamlet or Sherlock Holmes. So we need to have the option of including fictional characters in the UD. On the other hand, we never need to talk about every thing that does not exist. That might not even make sense. There are philosophical issues here that we will not try to address. We can avoid these difficulties by always specifying the UD. For example, if we mean to talk about plants, people, and cities, then the UD might be 'living things and places.'

Suppose that we want to translate sentence 20 and, with the same symbolization key, translate these sentences:

21. Esmerelda has a rose in her hair.
22. Everyone is cross with Esmerelda.

We need a UD that includes roses (so that we can symbolize sentence 20) and a UD that includes people (so we can translate sentence 21–22.) Here is a suitable key:

UD: people and plants
Px: x is a person.
Rx: x is a rose.
Tx: x has a thorn.
Cxy: x is cross with y.
Hxy: x has y in their hair.
e: Esmerelda

Since we do not have a predicate that means '... has a rose in her hair', translating sentence 21 will require paraphrasing. The sentence says that there is a rose in Esmerelda's hair; that is, there is something which is both a rose and is in Esmerelda's hair. So we get: $\exists x(Rx \wedge Hex)$.

It is tempting to translate sentence 22 as $\forall xCxe$. Unfortunately, this would mean that every member of the UD is cross with Esmerelda— both people and plants. It would mean, for instance, that the rose in Esmerelda's hair is cross with her. Of course, sentence 22 does not mean that.

'Everyone' means every person, not every member of the UD. So we can paraphrase sentence 22 as, 'Every person is cross with Esmerelda.' We know how to translate sentences like this: $\forall x(Px \rightarrow Cxe)$

In general, the universal quantifier can be used to mean 'everyone' if the UD contains only people. If there are people and other things in the UD, then 'everyone' must be treated as 'every person.'

Translating pronouns

When translating to QL, it is important to understand the structure of the sentences you want to translate. What matters is the final translation in QL, and sometimes you will be able to move from an English language sentence directly to a sentence of QL. Other times, it helps to paraphrase the sentence one or more times. Each successive paraphrase should move from the original sentence closer to something that you can translate directly into QL.

For the next several examples, we will use this symbolization key:

UD: people
Gx: x can play guitar.
Rx: x is a rock star.
l: Lemmy

Now consider these sentences:

23. If Lemmy can play guitar, then he is a rock star.
24. If a person can play guitar, then he is a rock star.

Sentence 23 and sentence 24 have the same consequent ('... he is a rock star'), but they cannot be translated in the same way. It helps to paraphrase the original sentences, replacing pronouns with explicit references.

Sentence 23 can be paraphrased as, 'If Lemmy can play guitar, then *Lemmy* is a rockstar.' This can obviously be translated as $Gl \rightarrow Rl$.

Sentence 24 must be paraphrased differently: 'If a person can play guitar, then *that person* is a rock star.' This sentence is not about any particular person, so we need a variable. Translating halfway, we can paraphrase the sentence as, 'For any person x, if x can play guitar, then x is a rockstar.' Now this can be translated as $\forall x(Gx \rightarrow Rx)$. This is the same as, 'Everyone who can play guitar is a rock star.'

Consider these further sentences:

25. If anyone can play guitar, then Lemmy can.
26. If anyone can play guitar, then he or she is a rock star.

These two sentences have the same antecedent ('If anyone can play guitar...'), but they have different logical structures.

Sentence 25 can be paraphrased, 'If someone can play guitar, then Lemmy can play guitar.' The antecedent and consequent are separate sentences, so it can be symbolized with a conditional as the main logical operator: $\exists xGx \rightarrow Gl$.

Sentence 26 can be paraphrased, 'For anyone, if that one can play guitar, then that one is a rock star.' It would be a mistake to symbolize this with an existential quantifier, because it is talking about everybody. The sentence is equivalent to 'All guitar players are rock stars.' It is best translated as $\forall x(Gx \rightarrow Rx)$.

The English words 'any' and 'anyone' should typically be translated using quantifiers. As these two examples show, they sometimes call for an existential quantifier (as in sentence 25) and sometimes for a universal quantifier (as in sentence 26). If you have a hard time determining which is required, paraphrase the sentence with an English language sentence that uses words besides 'any' or 'anyone.'

Quantifiers and scope

In the sentence $\exists xGx \rightarrow Gl$, the scope of the existential quantifier is the expression Gx. Would it matter if the scope of the quantifier were the whole sentence? That is, does the sentence $\exists x(Gx \rightarrow Gl)$ mean something different?

With the key given above, $\exists xGx \rightarrow Gl$ means that if there is some guitarist, then Lemmy is a guitarist. $\exists x(Gx \rightarrow Gl)$ would mean that there is some person such that if that person were a guitarist, then Lemmy would be a guitarist. Recall that the conditional here is a material conditional; the conditional is true if the antecedent is false. Let the constant p denote the author of this book, someone who is certainly not a guitarist. The sentence $Gp \rightarrow Gl$ is true because Gp is false. Since someone (namely p) satisfies the sentence, then $\exists x(Gx \rightarrow Gl)$ is true. The sentence is true because there is a non-guitarist, regardless of Lemmy's skill with the guitar.

Something strange happened when we changed the scope of the quantifier, because the conditional in QL is a material conditional. In order to keep the meaning the same, we would have to change the quantifier: $\exists x Gx \rightarrow Gl$ means the same thing as $\forall x(Gx \rightarrow Gl)$, and $\exists x(Gx \rightarrow Gl)$ means the same thing as $\forall x Gx \rightarrow Gl$.

This oddity does not arise with other connectives or if the variable is in the consequent of the conditional. For example, $\exists x Gx \wedge Gl$ means the same thing as $\exists x(Gx \wedge Gl)$, and $Gl \rightarrow \exists x Gx$ means the same things as $\exists x(Gl \rightarrow Gx)$.

Ambiguous predicates

Suppose we just want to translate this sentence:

27. Adina is a skilled surgeon.

Let the UD be people, let Kx mean 'x is a skilled surgeon', and let a mean Adina. Sentence 27 is simply Ka.

Suppose instead that we want to translate this argument:

> The hospital will only hire a skilled surgeon. All surgeons are greedy. Billy is a surgeon, but is not skilled. Therefore, Billy is greedy, but the hospital will not hire him.

We need to distinguish being a *skilled surgeon* from merely being a *surgeon*. So we define this symbolization key:

UD: people
Gx: x is greedy.
Hx: The hospital will hire x.
Rx: x is a surgeon.
Kx: x is skilled.
b: Billy

Now the argument can be translated in this way:

$$\forall x\big[\neg(Rx \wedge Kx) \rightarrow \neg Hx\big]$$
$$\forall x(Rx \rightarrow Gx)$$
$$Rb \wedge \neg Kb$$
$$\therefore\ Gb \wedge \neg Hb$$

Next suppose that we want to translate this argument:

> Carol is a skilled surgeon and a tennis player. Therefore, Carol is a skilled tennis player.

If we start with the symbolization key we used for the previous argument, we could add a predicate (let Tx mean 'x is a tennis player') and a constant (let c mean Carol). Then the argument becomes:

$$\begin{array}{ll} & (Rc \wedge Kc) \wedge Tc \\ \therefore & Tc \wedge Kc \end{array}$$

This translation is a disaster! It takes what in English is a terrible argument and translates it as a valid argument in QL. The problem is that there is a difference between being *skilled as a surgeon* and *skilled as a tennis player*. Translating this argument correctly requires two separate predicates, one for each type of skill. If we let K_1x mean 'x is skilled as a surgeon' and K_2x mean 'x is skilled as a tennis player,' then we can symbolize the argument in this way:

$$\begin{array}{ll} & (Rc \wedge K_1c) \wedge Tc \\ \therefore & Tc \wedge K_2c \end{array}$$

Like the English language argument it translates, this is invalid.

The moral of these examples is that you need to be careful of symbolizing predicates in an ambiguous way. Similar problems can arise with predicates like *good*, *bad*, *big*, and *small*. Just as skilled surgeons and skilled tennis players have different skills, big dogs, big mice, and big problems are big in different ways.

Is it enough to have a predicate that means 'x is a skilled surgeon', rather than two predicates 'x is skilled' and 'x is a surgeon'? Sometimes. As sentence 27 shows, sometimes we do not need to distinguish between skilled surgeons and other surgeons.

Must we always distinguish between different ways of being skilled, good, bad, or big? No. As the argument about Billy shows, sometimes we only need to talk about one kind of skill. If you are translating an argument that is just about dogs, it is fine to define a predicate that means 'x is big.' If the UD includes dogs and mice, however, it is probably best to make the predicate mean 'x is big for a dog.'

Multiple quantifiers

Consider this following symbolization key and the sentences that follow it:

UD: People and dogs

Dx: x is a dog.
Fxy: x is a friend of y.
Oxy: x owns y.
f: Fifi
g: Gerald

28. Fifi is a dog.
29. Gerald is a dog owner.
30. Someone is a dog owner.
31. All of Gerald's friends are dog owners.
32. Every dog owner is the friend of a dog owner.

Sentence 28 is easy: Df.

Sentence 29 can be paraphrased as, 'There is a dog that Gerald owns.' This can be translated as $\exists x(Dx \wedge Ogx)$.

Sentence 30 can be paraphrased as, 'There is some y such that y is a dog owner.' The subsentence 'y is a dog owner' is just like sentence 29, except that it is about y rather than being about Gerald. So we can translate sentence 30 as $\exists y \exists x(Dx \wedge Oyx)$.

Sentence 31 can be paraphrased as, 'Every friend of Gerald is a dog owner.' Translating part of this sentence, we get $\forall x(Fxg \rightarrow$ 'x is a dog owner'). Again, it is important to recognize that 'x is a dog owner' is structurally just like sentence 29. Since we already have an x-quantifier, we will need a different variable for the existential quantifier. Any other variable will do. Using z, sentence 31 can be translated as $\forall x\big[Fxg \rightarrow \exists z(Dz \wedge Oxz)\big]$.

Sentence 32 can be paraphrased as 'For any x that is a dog owner, there is a dog owner who is x's friend.' Partially translated, this becomes

$$\forall x\big[x \text{ is a dog owner} \rightarrow \exists y(y \text{ is a dog owner} \wedge Fxy)\big].$$

Completing the translation, sentence 32 becomes

$$\forall x\big[\exists z(Dz \wedge Oxz) \rightarrow \exists y\big(\exists z(Dz \wedge Oyz) \wedge Fxy\big)\big].$$

Consider this symbolization key and these sentences:

UD: people
Lxy: x likes y.
i: Imre.
k: Karl.

33. Imre likes everyone that Karl likes.
34. There is someone who likes everyone who likes everyone that he likes.

Sentence 33 can be partially translated as $\forall x$(Karl likes $x \rightarrow$ Imre likes x). This becomes $\forall x(Lkx \rightarrow Lix)$.

Sentence 34 is almost a tongue-twister. There is little hope of writing down the whole translation immediately, but we can proceed by small steps. An initial, partial translation might look like this:

$$\exists x \text{ everyone who likes everyone that } x \text{ likes is liked by } x$$

The part that remains in English is a universal sentence, so we translate further:

$$\exists x \forall y(y \text{ likes everyone that } x \text{ likes} \rightarrow x \text{ likes } y).$$

The antecedent of the conditional is structurally just like sentence 33, with y and x in place of Imre and Karl. So sentence 34 can be completely translated in this way

$$\exists x \forall y\big[\forall z(Lxz \rightarrow Lyz) \rightarrow Lxy\big]$$

When symbolizing sentences with multiple quantifiers, it is best to proceed by small steps. Paraphrase the English sentence so that the logical structure is readily symbolized in QL. Then translate piecemeal, replacing the daunting task of translating a long sentence with the simpler task of translating shorter formulae.

5.5 Sentences of QL

In this section, we provide a formal definition for a *well-formed formula* (wff) and *sentence* of QL.

Expressions

There are six kinds of symbols in QL:

predicates with subscripts, as needed	$A, B, C, \ldots, Z$ $A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$
constants with subscripts, as needed	$a, b, c, \ldots, w$ $a_1, w_4, h_7, m_{32}, \ldots$
variables with subscripts, as needed	x, y, z $x_1, y_1, z_1, x_2, \ldots$
connectives	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
parentheses	(,)
quantifiers	$\forall, \exists$

We define an EXPRESSION OF QL as any string of symbols of QL. Take any of the symbols of QL and write them down, in any order, and you have an expression.

Well-formed formulae

By definition, a TERM OF QL is either a constant or a variable.

An ATOMIC FORMULA OF QL is an n-place predicate followed by n terms.

Just as we did for SL, we will give a *recursive* definition for a wff of QL. In fact, most of the definition will look like the definition of for a wff of SL: Every atomic formula is a wff, and you can build new wffs by applying the sentential connectives.

We could just add a rule for each of the quantifiers and be done with it. For instance: If $\mathcal{A}$ is a wff, then $\forall x\mathcal{A}$ and $\exists x\mathcal{A}$ are wffs. However, this would allow for bizarre sentences like $\forall x\exists xDx$ and $\forall xDw$. What could these possibly mean? We could adopt some interpretation of such sentences, but instead we will write the definition of a wff so that such abominations do not even count as well-formed.

In order for $\forall x\mathcal{A}$ to be a wff, $\mathcal{A}$ must contain the variable x and must not already contain an x-quantifier. $\forall xDw$ will not count as a wff because 'x' does not occur in Dw, and $\forall x\exists xDx$ will not count as a wff because $\exists xDx$ contains an x-quantifier

1. Every atomic formula is a wff.
2. If $\mathcal{A}$ is a wff, then $\neg\mathcal{A}$ is a wff.
3. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, then $(\mathcal{A} \wedge \mathcal{B})$, is a wff.
4. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, $(\mathcal{A} \vee \mathcal{B})$ is a wff.
5. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, then $(\mathcal{A} \rightarrow \mathcal{B})$ is a wff.
6. If $\mathcal{A}$ and $\mathcal{B}$ are wffs, then $(\mathcal{A} \leftrightarrow \mathcal{B})$ is a wff.
7. If $\mathcal{A}$ is a wff, χ is a variable, $\mathcal{A}$ contains at least one occurrence of χ, and $\mathcal{A}$ contains no χ-quantifiers, then $\forall\chi\mathcal{A}$ is a wff.
8. If $\mathcal{A}$ is a wff, χ is a variable, $\mathcal{A}$ contains at least one occurrence of χ, and $\mathcal{A}$ contains no χ-quantifiers, then $\exists\chi\mathcal{A}$ is a wff.
9. All and only wffs of QL can be generated by applications of these rules.

Notice that the 'χ' that appears in the definition above is not the variable x. It is a *meta-variable* that stands in for any variable of QL. So $\forall xAx$ is a wff, but so are $\forall yAy$, $\forall zAz$, $\forall x_4Ax_4$, and $\forall z_9Az_9$.

We can now give a formal definition for scope: The SCOPE of a quantifier is the subformula for which the quantifier is the main logical operator.

Sentences

A sentence is something that can be either true or false. In SL, every wff was a sentence. This will not be the case in QL. Consider the following symbolization key:

UD: people
Lxy: x loves y
b: Boris

Consider the expression Lzz. It is an atomic forumula: a two-place predicate followed by two terms. All atomic formula are wffs, so Lzz is a wff. Does it mean anything? You might think that it means that z loves himself, in the same way that Lbb means that Boris loves himself. Yet z is a variable; it does not name some person the way a constant would. The wff Lzz does not tell us how to interpret z. Does it mean everyone? anyone? someone? If we had a z-quantifier, it would tell us how to interpret z. For instance, $\exists zLzz$ would mean that someone loves themselves.

Some formal languages treat a wff like Lzz as implicitly having a universal quantifier in front. We will not do this for QL. If you mean to say that everyone loves themself, then you need to write the quantifier: $\forall zLzz$

In order to make sense of a variable, we need a quantifier to tell us how to interpret that variable. The scope of an x-quantifier, for instance, is the part of the formula where the quantifier tells how to interpret x.

In order to be precise about this, we define a BOUND VARIABLE to be an occurrence of a variable χ that is within the scope of an χ-quantifier. A FREE VARIABLE is an occurance of a variable that is not bound.

For example, consider the wff $\forall x(Ex \vee Dy) \rightarrow \exists z(Ex \rightarrow Lzx)$. The scope of the universal quantifier $\forall x$ is $(Ex \vee Dy)$, so the first x is bound by the universal quantifier but the second and third xs are free. There is not y-quantifier, so the y is free. The scope of the existential quantifier $\exists z$ is $(Ex \rightarrow Lzx)$, so both occurrences of z are bound by it.

We define a SENTENCE of QL as a wff of QL that contains no free variables.

Notational conventions

We will adopt the same notational conventions that we did for SL (p. 30.) First, we may leave off the outermost parentheses of a formula. Second, we will use square brackets '[' and ']' in place of parentheses to increase the readability of formulae. Third, we will leave out parentheses between each pair of conjuncts when writing long series of conjunctions. Fourth, we will leave out parentheses between each pair of disjuncts when writing long series of disjunctions.

5.6 Identity

Consider this sentence:

35. Pavel owes money to everyone else.

Let the UD be people; this will allow us to translate 'everyone' as a universal quantifier. Let Oxy mean 'x owes money to y', and let p mean Pavel. Now we can symbolize sentence 35 as $\forall x Opx$. Unfortunately, this translation has some odd consequences. It says that Pavel owes money to every member of the UD, including Pavel; it entails that Pavel owes money to himself. However, sentence 35 does not say that Pavel owes money to himself; he owes money to everyone *else*. This is a problem, because $\forall x Opx$ is the best translation we can give of this sentence into QL.

The solution is to add another symbol to QL. The symbol '=' is a two-place predicate. Since it has a special logical meaning, we write it a bit differently: For two terms t_1 and t_2, $t_1 = t_2$ is an atomic formula.

The predicate $x = y$ means 'x is identical to y.' This does not mean merely that x and y are indistinguishable or that all of the same predicates are true of them. Rather, it means that x and y are the very same thing.

When we write $x \neq y$, we mean that x and y are not identical. There is no reason to introduce this as an additional predicate. Instead, $x \neq y$ is an abbreviation of $\neg(x = y)$.

Now suppose we want to symbolize this sentence:

36. Pavel is Mister Checkov.

Let the constant c mean Mister Checkov. Sentence 36 can be symbolized as $p = c$. This means that the constants p and c both refer to the same guy.

This is all well and good, but how does it help with sentence 35? That sentence can be paraphrased as, 'Everyone who is not Pavel is owed money by Pavel.' This is a sentence structure we already know how to symbolize: 'For all x, if x is not Pavel, then x is owed money by Pavel.' In QL with identity, this becomes $\forall x(x \neq p \rightarrow Opx)$.

In addition to sentences that use the word 'else', identity will be helpful when symbolizing some sentences that contain the words 'besides' and 'only.' Consider these examples:

37. No one besides Pavel owes money to Hikaru.
38. Only Pavel owes Hikaru money.

We add the constant h, which means Hikaru.

Sentence 37 can be paraphrased as, 'No one who is not Pavel owes money to Hikaru.' This can be translated as $\neg\exists x(x \neq p \wedge Oxh)$.

Sentence 38 can be paraphrased as, 'Pavel owes Hikaru *and* no one besides Pavel owes Hikaru money.' We have already translated one of the conjuncts, and the other is straightforward. Sentence 38 becomes $Oph \wedge \neg\exists x(x \neq p \wedge Oxh)$.

Expressions of quantity

We can also use identity to say how many things there are of a particular kind. For example, consider these sentences:

39. There is at least one apple on the table.
40. There are at least two apples on the table.
41. There are at least three apples on the table.

Let the UD be *things on the table*, and let Ax mean 'x is an apple.'

Sentence 39 does not require identity. It can be translated adequately as $\exists x Ax$: There is some apple on the table— perhaps many, but at least one.

It might be tempting to also translate sentence 40 without identity. Yet consider the sentence $\exists x\exists y(Ax \wedge Ay)$. It means that there is some apple x in the UD and some apple y in the UD. Since nothing precludes x and y from picking out the same member of the UD, this would be true even if there were only one apple. In order to make sure that there are two *different* apples, we need an identity predicate. Sentence 40 needs to say that the two apples that exist are not identical, so it can be translated as $\exists x\exists y(Ax \wedge Ay \wedge x \neq y)$.

Sentence 41 requires talking about three different apples. It can be translated as $\exists x\exists y\exists z(Ax \wedge Ay \wedge Az \wedge x \neq y \wedge y \neq z \wedge x \neq z)$.

Continuing in this way, we could translate 'There are at least n apples on the table.' There is a summary of how to symbolize sentences like these on p. 148.

Now consider these sentences:

42. There is at most one apple on the table.
43. There are at most two apples on the table.

Sentence 42 can be paraphrased as, 'It is not the case that there are at least *two* apples on the table.' This is just the negation of sentence 40:

$$\neg\exists x\exists y(Ax \wedge Ay \wedge x \neq y)$$

Sentence 42 can also be approached in another way. It means that any apples that there are on the table must be the selfsame apple, so it can be translated as $\forall x \forall y\big[(Ax \wedge Ay) \rightarrow x = y\big]$. The two translations are logically equivalent, so both are correct.

In a similar way, sentence 43 can be translated in two equivalent ways. It can be paraphrased as, 'It is not the case that there are *three* or more distinct apples', so it can be translated as the negation of sentence 41. Using universal quantifiers, it can also be translated as

$$\forall x \forall y \forall z\big[(Ax \wedge Ay \wedge Az) \rightarrow (x = y \vee x = z \vee y = z)\big].$$

See p. 148 for the general case.

The examples above are sentences about apples, but the logical structure of the sentences translates mathematical inequalities like $a \geq 3$, $a \leq 2$, and so on. We also want to be able to translate statements of equality which say exactly how many things there are. For example:

44. There is exactly one apple on the table.
45. There are exactly two apples on the table.

Sentence 44 can be paraphrased as, 'There is *at least* one apple on the table, and there is *at most* one apple on the table.' This is just the conjunction of sentence 39 and sentence 42: $\exists x Ax \wedge \forall x \forall y\big[(Ax \wedge Ay) \rightarrow x = y\big]$. This is a somewhat complicated way of going about it. It is perhaps more straightforward to paraphrase sentence 44 as, 'There is a thing which is the only apple on the table.' Thought of in this way, the sentence can be translated $\exists x\big[Ax \wedge \neg\exists y(Ay \wedge x \neq y)\big]$.

Similarly, sentence 45 may be paraphrased as, 'There are two different apples on the table, and these are the only apples on the table.' This can be translated as $\exists x \exists y\big[Ax \wedge Ay \wedge x \neq y \wedge \neg\exists z(Az \wedge x \neq z \wedge y \neq z)\big]$.

Finally, consider this sentence:

46. There are at most two things on the table.

It might be tempting to add a predicate so that Tx would mean 'x is a thing on the table.' However, this is unnecessary. Since the UD is the set of things on the table, all members of the UD are on the table. If we want to talk about a *thing on the table*, we need only use a quantifier. Sentence 46 can be symbolized like sentence 43 (which said that there were at most two apples), but leaving out the predicate entirely. That is, sentence 46 can be translated as $\forall x \forall y \forall z(x = y \vee x = z \vee y = z)$.

Techniques for symbolizing expressions of quantity ('at most', 'at least', and 'exactly') are summarized on p. 148.

Definite descriptions

Recall that a constant of QL must refer to some member of the UD. This constraint allows us to avoid the problem of non-referring terms. Given a UD that included only actually existing creatures but a constant c that meant 'chimera' (a mythical creature), sentences containing c would become impossible to evaluate.

The most widely influential solution to this problem was introduced by Bertrand Russell in 1905. Russell asked how we should understand this sentence:

47. The present king of France is bald.

The phrase 'the present king of France' is supposed to pick out an individual by means of a definite description. However, there was no king of France in 1905 and there is none now. Since the description is a non-referring term, we cannot just define a constant to mean 'the present king of France' and translate the sentence as Kf.

Russell's idea was that sentences that contain definite descriptions have a different logical structure than sentences that contain proper names, even though they share the same grammatical form. What do we mean when we use an unproblematic, referring description, like 'the highest peak in Washington state'? We mean that there is such a peak, because we could not talk about it otherwise. We also mean that it is the only such peak. If there was another peak in Washington state of exactly the same height as Mount Rainier, then Mount Rainier would not be *the* highest peak.

According to this analysis, sentence 47 is saying three things. First, it makes an *existence* claim: There is some present king of France. Second, it makes a *uniqueness* claim: This guy is the only present king of France. Third, it makes a claim of *predication*: This guy is bald.

In order to symbolize definite descriptions in this way, we need the identity predicate. Without it, we could not translate the uniqueness claim which (according to Russell) is implicit in the definite description.

Let the UD be *people actually living*, let Fx mean 'x is the present king of France', and let Bx mean 'x is bald.' Sentence 47 can then be translated as $\exists x\big[Fx \wedge \neg\exists y(Fy \wedge x \neq y) \wedge Bx\big]$. This says that there is some guy who is the present king of France, he is the only present king of France, and he is bald.

Understood in this way, sentence 47 is meaningful but false. It says that this guy exists, but he does not.

The problem of non-referring terms is most vexing when we try to translate negations. So consider this sentence:

48. The present king of France is not bald.

According to Russell, this sentence is ambiguous in English. It could mean either of two things:

48a. It is not the case that the present king of France is bald.
48b. The present king of France is non-bald.

Both possible meanings negate sentence 47, but they put the negation in different places.

Sentence 48a is called a WIDE-SCOPE NEGATION, because it negates the entire sentence. It can be translated as $\neg\exists x\big[Fx \wedge \neg\exists y(Fy \wedge x \neq y) \wedge Bx\big]$. This does not say anything about the present king of France, but rather says that some sentence about the present king of France is false. Since sentence 47 if false, sentence 48a is true.

Sentence 48b says something about the present king of France. It says that he lacks the property of baldness. Like sentence 47, it makes an existence claim and a uniqueness claim; it just denies the claim of predication. This is called NARROW-SCOPE NEGATION. It can be translated as $\exists x\big[Fx \wedge \neg\exists y(Fy \wedge x \neq y) \wedge \neg Bx\big]$. Since there is no present king of France, this sentence is false.

Russell's theory of definite descriptions resolves the problem of non-referring terms and also explains why it seemed so paradoxical. Before we distinguished between the wide-scope and narrow-scope negations, it seemed that sentences like 48 should be both true and false. By showing that such sentences are ambiguous, Russell showed that they are true understood one way but false understood another way.

For a more detailed discussion of Russell's theory of definite descriptions, including objections to it, see Peter Ludlow's entry 'descriptions' in *The Stanford Encyclopedia of Philosophy*: Summer 2005 edition, edited by Edward N. Zalta, `http://plato.stanford.edu/archives/sum2005/entries/descriptions/`

Practice Exercises

⋆ **Part A** Using the symbolization key given, translate each English-language sentence into QL.

UD: all animals
Ax: x is an alligator.
Mx: x is a monkey.
Rx: x is a reptile.
Zx: x lives at the zoo.

Lxy: x loves y.
a: Amos
b: Bouncer
c: Cleo

1. Amos, Bouncer, and Cleo all live at the zoo.
2. Bouncer is a reptile, but not an alligator.
3. If Cleo loves Bouncer, then Bouncer is a monkey.
4. If both Bouncer and Cleo are alligators, then Amos loves them both.
5. Some reptile lives at the zoo.
6. Every alligator is a reptile.
7. Any animal that lives at the zoo is either a monkey or an alligator.
8. There are reptiles which are not alligators.
9. Cleo loves a reptile.
10. Bouncer loves all the monkeys that live at the zoo.
11. All the monkeys that Amos loves love him back.
12. If any animal is a reptile, then Amos is.
13. If any animal is an alligator, then it is a reptile.
14. Every monkey that Cleo loves is also loved by Amos.
15. There is a monkey that loves Bouncer, but sadly Bouncer does not reciprocate this love.

Part B These are syllogistic figures identified by Aristotle and his successors, along with their medieval names. Translate each argument into QL.

1. **Barbara** All Bs are Cs. All As are Bs. $\therefore$ All As are Cs.
2. **Baroco** All Cs are Bs. Some A is not B. $\therefore$ Some A is not C.
3. **Bocardo** Some B is not C. All Bs are As. $\therefore$ Some A is not C.
4. **Celantes** No Bs are Cs. All As are Bs. $\therefore$ No Cs are As.
5. **Celarent** No Bs are Cs. All As are Bs. $\therefore$ No As are Cs.
6. **Camestres** All Cs are Bs. No As are Bs. $\therefore$ No As are Cs.
7. **Cesare** No Cs are Bs. All As are Bs. $\therefore$ No As are Cs.
8. **Dabitis** All Bs are Cs. Some A is B. $\therefore$ Some C is A.
9. **Darii** All Bs are Cs. Some A is B. $\therefore$ Some A is C.
10. **Datisi** All Bs are Cs. Some B is A. $\therefore$ Some A is C.
11. **Disamis** Some A is B. All As are Cs. $\therefore$ Some B is C.
12. **Ferison** No Bs are Cs. Some A is B. $\therefore$ Some A is not C.
13. **Ferio** No Bs are Cs. Some B is A. $\therefore$ Some A is not C.
14. **Festino** No Cs are Bs. Some A is B. $\therefore$ Some A is not C.
15. **Baralipton** All Bs are Cs. All As are Bs. $\therefore$ Some C is A.
16. **Frisesomorum** Some B is C. No As are Bs. $\therefore$ Some C is not A.

Part C Using the symbolization key given, translate each English-language sentence into QL.

UD: all animals

Dx: x is a dog.
Sx: x likes samurai movies.
Lxy: x is larger than y.
b: Bertie
e: Emerson
f: Fergis

1. Bertie is a dog who likes samurai movies.
2. Bertie, Emerson, and Fergis are all dogs.
3. Emerson is larger than Bertie, and Fergis is larger than Emerson.
4. All dogs like samurai movies.
5. Only dogs like samurai movies.
6. There is a dog that is larger than Emerson.
7. If there is a dog larger than Fergis, then there is a dog larger than Emerson.
8. No animal that likes samurai movies is larger than Emerson.
9. No dog is larger than Fergis.
10. Any animal that dislikes samurai movies is larger than Bertie.
11. There is an animal that is between Bertie and Emerson in size.
12. There is no dog that is between Bertie and Emerson in size.
13. No dog is larger than itself.
14. For every dog, there is some dog larger than it.
15. There is an animal that is smaller than every dog.
16. If there is an animal that is larger than any dog, then that animal does not like samurai movies.

Part D For each argument, write a symbolization key and translate the argument into QL.

1. Nothing on my desk escapes my attention. There is a computer on my desk. As such, there is a computer that does not escape my attention.
2. All my dreams are black and white. Old TV shows are in black and white. Therefore, some of my dreams are old TV shows.
3. Neither Holmes nor Watson has been to Australia. A person could see a kangaroo only if they had been to Australia or to a zoo. Although Watson has not seen a kangaroo, Holmes has. Therefore, Holmes has been to a zoo.
4. No one expects the Spanish Inquisition. No one knows the troubles I've seen. Therefore, anyone who expects the Spanish Inquisition knows the troubles I've seen.
5. An antelope is bigger than a bread box. I am thinking of something that is no bigger than a bread box, and it is either an antelope or a cantaloupe. As such, I am thinking of a cantaloupe.
6. All babies are illogical. Nobody who is illogical can manage a crocodile. Berthold is a baby. Therefore, Berthold is unable to manage a crocodile.

$\star$ **Part E** Using the symbolization key given, translate each English-language sentence into QL.

UD: candies
Cx: x has chocolate in it.
Mx: x has marzipan in it.
Sx: x has sugar in it.
Tx: Boris has tried x.
Bxy: x is better than y.

1. Boris has never tried any candy.
2. Marzipan is always made with sugar.
3. Some candy is sugar-free.
4. The very best candy is chocolate.
5. No candy is better than itself.
6. Boris has never tried sugar-free chocolate.
7. Boris has tried marzipan and chocolate, but never together.
8. Any candy with chocolate is better than any candy without it.
9. Any candy with chocolate and marzipan is better than any candy that lacks both.

Part F Using the symbolization key given, translate each English-language sentence into QL.

UD: people and dishes at a potluck
Rx: x has run out.
Tx: x is on the table.
Fx: x is food.
Px: x is a person.
Lxy: x likes y.
e: Eli
f: Francesca
g: the guacamole

1. All the food is on the table.
2. If the guacamole has not run out, then it is on the table.
3. Everyone likes the guacamole.
4. If anyone likes the guacamole, then Eli does.
5. Francesca only likes the dishes that have run out.
6. Francesca likes no one, and no one likes Francesca.
7. Eli likes anyone who likes the guacamole.
8. Eli likes anyone who likes the people that he likes.
9. If there is a person on the table already, then all of the food must have run out.

⋆ **Part G** Using the symbolization key given, translate each English-language sentence into QL.

UD: people

Dx: x dances ballet.
Fx: x is female.
Mx: x is male.
Cxy: x is a child of y.
Sxy: x is a sibling of y.
e: Elmer
j: Jane
p: Patrick

1. All of Patrick's children are ballet dancers.
2. Jane is Patrick's daughter.
3. Patrick has a daughter.
4. Jane is an only child.
5. All of Patrick's daughters dance ballet.
6. Patrick has no sons.
7. Jane is Elmer's niece.
8. Patrick is Elmer's brother.
9. Patrick's brothers have no children.
10. Jane is an aunt.
11. Everyone who dances ballet has a sister who also dances ballet.
12. Every man who dances ballet is the child of someone who dances ballet.

Part H Identify which variables are bound and which are free.

1. $\exists xLxy \wedge \forall yLyx$
2. $\forall xAx \wedge Bx$
3. $\forall x(Ax \wedge Bx) \wedge \forall y(Cx \wedge Dy)$
4. $\forall x\exists y[Rxy \rightarrow (Jz \wedge Kx)] \vee Ryx$
5. $\forall x_1(Mx_2 \leftrightarrow Lx_2x_1) \wedge \exists x_2Lx_3x_2$

Part I Using the symbolization key given, translate each English-language sentence into QL with identity. The last sentence is ambiguous and can be translated two ways; you should provide both translations. (Hint: Identity is only required for the last four sentences.)

UD: people
Kx: x knows the combination to the safe.
Sx: x is a spy.
Vx: x is a vegetarian.
Txy: x trusts y.
h: Hofthor
i: Ingmar

1. Hofthor is a spy, but no vegetarian is a spy.
2. No one knows the combination to the safe unless Ingmar does.
3. No spy knows the combination to the safe.

4. Neither Hofthor nor Ingmar is a vegetarian.
5. Hofthor trusts a vegetarian.
6. Everyone who trusts Ingmar trusts a vegetarian.
7. Everyone who trusts Ingmar trusts someone who trusts a vegetarian.
8. Only Ingmar knows the combination to the safe.
9. Ingmar trusts Hofthor, but no one else.
10. The person who knows the combination to the safe is a vegetarian.
11. The person who knows the combination to the safe is not a spy.

⋆ **Part J** Using the symbolization key given, translate each English-language sentence into QL with identity. The last two sentences are ambiguous and can be translated two ways; you should provide both translations for each.

UD: cards in a standard deck
Bx: x is black.
Cx: x is a club.
Dx: x is a deuce.
Jx: x is a jack.
Mx: x is a man with an axe.
Ox: x is one-eyed.
Wx: x is wild.

1. All clubs are black cards.
2. There are no wild cards.
3. There are at least two clubs.
4. There is more than one one-eyed jack.
5. There are at most two one-eyed jacks.
6. There are two black jacks.
7. There are four deuces.
8. The deuce of clubs is a black card.
9. One-eyed jacks and the man with the axe are wild.
10. If the deuce of clubs is wild, then there is exactly one wild card.
11. The man with the axe is not a jack.
12. The deuce of clubs is not the man with the axe.

Part K Using the symbolization key given, translate each English-language sentence into QL with identity. The last two sentences are ambiguous and can be translated two ways; you should provide both translations for each.

UD: animals in the world
Bx: x is in Farmer Brown's field.
Hx: x is a horse.
Px: x is a Pegasus.
Wx: x has wings.

1. There are at least three horses in the world.

2. There are at least three animals in the world.
3. There is more than one horse in Farmer Brown's field.
4. There are three horses in Farmer Brown's field.
5. There is a single winged creature in Farmer Brown's field; any other creatures in the field must be wingless.
6. The Pegasus is a winged horse.
7. The animal in Farmer Brown's field is not a horse.
8. The horse in Farmer Brown's field does not have wings.

Chapter 6

Formal semantics

In this chapter, we describe a *formal semantics* for SL and for QL. The word 'semantics' comes from the greek word for 'mark' and means 'related to meaning.' So a formal semantics will be a mathematical account of meaning in the formal language.

A formal, logical language is built from two kinds of elements: logical symbols and non-logical symbols. Connectives (like '$\wedge$') and quantifiers (like '$\forall$') are logical symbols, because their meaning is specified within the formal language. When writing a symbolization key, you are not allowed to change the meaning of the logical symbols. You cannot say, for instance, that the '$\neg$' symbol will mean 'not' in one argument and 'perhaps' in another. The '$\neg$' symbol always means logical negation. It is used to translate the English language word 'not', but it is a symbol of a formal language and is defined by its truth conditions.

The sentence letters in SL are non-logical symbols, because their meaning is not defined by the logical structure of SL. When we translate an argument from English to SL, for example, the sentence letter M does not have its meaning fixed in advance; instead, we provide a symbolization key that says how M should be interpreted in that argument. In QL, the predicates and constants are non-logical symbols.

In translating from English to a formal language, we provided symbolization keys which were interpretations of all the non-logical symbols we used in the translation. An INTERPRETATION gives a meaning to all the non-logical elements of the language.

It is possible to provide different interpretations that make no formal difference. In SL, for example, we might say that D means 'Today is Tuesday'; we might say instead that D means 'Today is the day after Monday.' These are two different interpretations, because they use different English sentences for the meaning of D. Yet, formally, there is no difference between them. All that matters once we have symbolized these sentences is whether they are true or false. In order to characterize what makes a difference in the formal language, we need to know what makes sentences true or false. For this, we need a formal characterization of *truth*.

When we gave definitions for a sentence of SL and for a sentence of QL, we distinguished between the OBJECT LANGUAGE and the METALANGUAGE. The object language is the language that we are *talking about*: either SL or QL. The metalanguage is the language that we use to talk about the object language: English, supplemented with some mathematical jargon. It will be important to keep this distinction in mind.

6.1 Semantics for SL

This section provides a rigorous, formal characterization of *truth in SL* which builds on what we already know from doing truth tables. We were able to use truth tables to reliably test whether a sentence was a tautology in SL, whether two sentences were equivalent, whether an argument was valid, and so on. For instance: $\mathcal{A}$ is a tautology in SL if it is T on every line of a complete truth table.

This worked because each line of a truth table corresponds to a way the world might be. We considered all the possible combinations of 1 and 0 for the sentence letters that made a difference to the sentences we cared about. The truth table allowed us to determine what would happen given these different combinations.

Once we construct a truth table, the symbols '1' and '0' are divorced from their metalinguistic meaning of 'true' and 'false'. We interpret '1' as meaning 'true', but the formal properties of 1 are defined by the characteristic truth tables for the various connectives. The symbols in a truth table have a formal meaning that we can specific entirely in terms of how the connectives operate. For example, if A is value 1, then $\neg A$ is value 0.

In short: Truth in SL just is the assignment of a 1 or a 0.

To formally define truth in SL, then, we want a function that assigns a 1 or 0 to each of the sentences of SL. We can interpret this function as a definition of truth for SL if it assigns 1 to all of the true sentences of SL and 0 to all of the false sentences of SL. Call this function 'v' (for 'valuation'). We want v to be a function such that for any sentence $\mathcal{A}$, $v(\mathcal{A}) = 1$ if $\mathcal{A}$ is true and $v(\mathcal{A}) = 0$ if $\mathcal{A}$ is false.

Recall that the recursive definition of a wff for SL had two stages: The first step said that atomic sentences (solitary sentence letters) are wffs. The second stage allowed for wffs to be constructed out of more basic wffs. There were clauses of the definition for all of the sentential connectives. For example, if $\mathcal{A}$ is a wff, then $\neg\mathcal{A}$ is a wff.

Our strategy for defining the truth function, v, will also be in two steps. The first step will handle truth for atomic sentences; the second step will handle truth for compound sentences.

Truth in SL

How can we define truth for an atomic sentence of SL? Consider, for example, the sentence M. Without an interpretation, we cannot say whether M is true or false. It might mean anything. If we use M to symbolize 'The moon orbits the Earth', then M is true. If we use M to symbolize 'The moon is a giant turnip', then M is false.

Moreover, the way you would discover whether or not M is true depends on what M means. If M means 'It is Monday,' then you would need to check a calendar. If M means 'Jupiter's moon Io has significant volcanic activity,' then you would need to check an astronomy text— and astronomers know because they sent satellites to observe Io.

When we give a symbolization key for SL, we provide an interpretation of the sentence letters that we use. The key gives an English language sentence for each sentence letter that we use. In this way, the interpretation specifies what each of the sentence letters *means*. However, this is not enough to determine whether or not that sentence is true. The sentences about the moon, for instance, require that you know some rudimentary astronomy. Imagine a small child who became convinced that the moon is a giant turnip. She could understand what the sentence 'The moon is a giant turnip' means, but mistakenly think that it was true.

Consider another example: If M means 'It is morning now', then whether it is true or not depends on when you are reading this. I know what the sentence means, but— since I do not know when you will be reading this— I do not know whether it is true or false.

So an interpretation alone does not determine whether a sentence is true or false. Truth or falsity depends also on what the world is like. If M meant 'The moon is a giant turnip' and the real moon were a giant turnip, then M would be true. To put the point in a general way, truth or falsity is determined by an interpretation *plus* a way that the world is.

$$\text{INTERPRETATION} + \text{STATE OF THE WORLD} \Longrightarrow \text{TRUTH/FALSITY}$$

In providing a logical definition of truth, we will not be able to give an account of how an atomic sentence is made true or false by the world. Instead, we will introduce a *truth value assignment*. Formally, this will be a function that tells us the truth value of all the atomic sentences. Call this function 'a' (for 'assignment'). We define a for all sentence letters $\mathcal{P}$, such that

$$a(\mathcal{P}) = \begin{cases} 1 & \text{if } \mathcal{P} \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

This means that a takes any sentence of SL and assigns it either a one or a zero; one if the sentence is true, zero if the sentence is false. The details of the function a are determined by the meaning of the sentence letters together with the state of the world. If D means 'It is dark outside', then $a(D) = 1$ at night or during a heavy storm, while $a(D) = 0$ on a clear day.

You can think of a as being like a row of a truth table. Whereas a truth table row assigns a truth value to a few atomic sentences, the truth value assignment assigns a value to every atomic sentence of SL. There are infinitely many sentence letters, and the truth value assignment gives a value to each of them. When constructing a truth table, we only care about sentence letters that affect the truth value of sentences that interest us. As such, we ignore the rest. Strictly speaking, every row of a truth table gives a *partial* truth value assignment.

It is important to note that the truth value assignment, a, is not part of the language SL. Rather, it is part of the mathematical machinery that we are using to describe SL. It encodes which atomic sentences are true and which are false.

We now define the truth function, v, using the same recursive structure that we used to define a wff of SL.

1. If $\mathcal{A}$ is a sentence letter, then $v(\mathcal{A}) = a(\mathcal{A})$.
2. If $\mathcal{A}$ is $\neg\mathcal{B}$ for some sentence $\mathcal{B}$, then
$$v(\mathcal{A}) = \begin{cases} 1 & \text{if } v(\mathcal{B}) = 0, \\ 0 & \text{otherwise.} \end{cases}$$
3. If $\mathcal{A}$ is $(\mathcal{B} \wedge \mathcal{C})$ for some sentences $\mathcal{B}, \mathcal{C}$, then
$$v(\mathcal{A}) = \begin{cases} 1 & \text{if } v(\mathcal{B}) = 1 \text{ and } v(\mathcal{C}) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

It might seem as if this definition is circular, because it uses the word 'and' in trying to define 'and.' Notice, however, that this is not a definition of the English word 'and'; it is a definition of truth for sentences of SL containing the logical symbol '$\wedge$.' We define truth for object language sentences containing the symbol '$\wedge$' using the metalanguage word 'and.' There is nothing circular about that.

4. If $\mathcal{A}$ is $(\mathcal{B} \vee \mathcal{C})$ for some sentences $\mathcal{B}, \mathcal{C}$, then
$$v(\mathcal{A}) = \begin{cases} 0 & \text{if } v(\mathcal{B}) = 0 \text{ and } v(\mathcal{C}) = 0, \\ 1 & \text{otherwise.} \end{cases}$$

5. If $\mathcal{A}$ is $(\mathcal{B} \to \mathcal{C})$ for some sentences $\mathcal{B}, \mathcal{C}$, then

$$v(\mathcal{A}) = \begin{cases} 0 & \text{if } v(\mathcal{B}) = 1 \text{ and } v(\mathcal{C}) = 0, \\ 1 & \text{otherwise.} \end{cases}$$

6. If $\mathcal{A}$ is $(\mathcal{B} \leftrightarrow \mathcal{C})$ for some sentences $\mathcal{B}, \mathcal{C}$, then

$$v(\mathcal{A}) = \begin{cases} 1 & \text{if } v(\mathcal{B}) = v(\mathcal{C}), \\ 0 & \text{otherwise.} \end{cases}$$

Since the definition of v has the same structure as the definition of a wff, we know that v assigns a value to *every* wff of SL. Since the sentences of SL and the wffs of SL are the same, this means that v returns the truth value of every sentence of SL.

Truth in SL is always truth *relative to* some truth value assignment, because the definition of truth for SL does not say whether a given sentence is true or false. Rather, it says how the truth of that sentence relates to a truth value assignment.

Other concepts in SL

Working with SL so far, we have done without a precise definition of 'tautology', 'contradiction', and so on. Truth tables provided a way to *check if* a sentence was a tautology in SL, but they did not *define* what it means to be a tautology in SL. We will give definitions of these concepts for SL in terms of entailment.

The relation of semantic entailment, '$\mathcal{A}$ entails $\mathcal{B}$', means that there is no truth value assignment for which $\mathcal{A}$ is true and $\mathcal{B}$ is false. Put differently, it means that $\mathcal{B}$ is true for any and all truth value assignments for which $\mathcal{A}$ is true.

We abbreviate this with a symbol called the *double turnstile*: $\mathcal{A} \models \mathcal{B}$ means '$\mathcal{A}$ semantically entails $\mathcal{B}$.'

We can talk about entailment between more than two sentences:

$$\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\} \models \mathcal{B}$$

means that there is no truth value assignment for which all of the sentences in the set $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\}$ are true and $\mathcal{B}$ is false.

We can also use the symbol with just one sentence: $\models \mathcal{C}$ means that $\mathcal{C}$ is true for all truth value assignments. This is equivalent to saying that the sentence is entailed by anything.

The double turnstile symbol allows us to give concise definitions for various concepts of SL:

A TAUTOLOGY IN SL is a sentence $\mathcal{A}$ such that $\models \mathcal{A}$.

A CONTRADICTION IN SL is a sentence $\mathcal{A}$ such that $\models \neg\mathcal{A}$.

A sentence is CONTINGENT IN SL if and only if it is neither a tautology nor a contradiction.

An argument " $\mathcal{P}_1, \mathcal{P}_2, \cdots, \therefore \mathcal{C}$ " is VALID IN SL if and only if $\{\mathcal{P}_1, \mathcal{P}_2, \cdots\} \models \mathcal{C}$.

Two sentences $\mathcal{A}$ and $\mathcal{B}$ are LOGICALLY EQUIVALENT IN SL if and only if both $\mathcal{A} \models \mathcal{B}$ and $\mathcal{B} \models \mathcal{A}$.

Logical consistency is somewhat harder to define in terms of semantic entailment. Instead, we will define it in this way:

The set $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\}$ is CONSISTENT IN SL if and only if there is at least one truth value assignment for which all of the sentences are true. The set is INCONSISTENT IN SL if and if only there is no such assignment.

6.2 Interpretations and models in QL

In SL, an interpretation or symbolization key specifies what each of the sentence letters means. The interpretation of a sentence letter along with the state of the world determines whether the sentence letter is true or false. Since the basic units are sentence letters, an interpretation only matters insofar as it makes sentence letters true or false. Formally, the semantics for SL is strictly in terms of truth value assignments. Two interpretations are the same, formally, if they make for the same truth value assignment.

What is an interpretation in QL? Like a symbolization key for QL, an interpretation requires a UD, a schematic meaning for each of the predicates, and an object that is picked out by each constant. For example:

UD: comic book characters
Fx: x fights crime.
b: the Batman
w: Bruce Wayne

Consider the sentence Fb. The sentence is true on this interpretation, but— just as in SL— the sentence is not true *just because* of the interpretation. Most people in our culture know that Batman fights crime, but this requires a modicum of knowledge about comic books. The sentence Fb is true because of the interpretation *plus* some facts about comic books. This is especially obvious when we consider Fw. Bruce Wayne is the secret identity of the Batman in the comic books— the identity claim $b = w$ is true— so Fw is true. Since it is a *secret* identity, however, other characters do not know that Fw is true even though they know that Fb is true.

We could try to characterize this as a truth value assignment, as we did for SL. The truth value assignment would assign 0 or 1 to each atomic wff: Fb, Fw, and so on. If we were to do that, however, we might just as well translate the sentences from QL to SL by replacing Fb and Fw with sentence letters. We could then rely on the definition of truth for SL, but at the cost of ignoring all the logical structure of predicates and terms. In writing a symbolization key for QL, we do not give separate definitions for Fb and Fw. Instead, we give meanings to F, b, and w. This is essential because we want to be able to use quantifiers. There is no adequate way to translate $\forall xFx$ into SL.

So we want a formal counterpart to an interpretation for predicates and constants, not just for sentences. We cannot use a truth value assignment for this, because a predicate is neither true nor false. In the interpretation given above, F is true *of* the Batman (i.e., Fb is true), but it makes no sense at all to ask whether F on its own is true. It would be like asking whether the English language fragment '. . .fights crime' is true.

What does an interpretation do for a predicate, if it does not make it true or false? An interpretation helps to pick out the objects to which the predicate applies. Interpreting Fx to mean 'x fights crime' picks out Batman, Superman, Spiderman, and other heroes as the things that are Fs. Formally, this is a set of members of the UD to which the predicate applies; this set is called the EXTENSION of the predicate.

Many predicates have indefinitely large extensions. It would be impractical to try and write down all of the comic book crime fighters individually, so instead we use an English language expression to interpret the predicate. This is somewhat imprecise, because the interpretation alone does not tell you which members of the UD are in the extension of the predicate. In order to figure out whether a particular member of the UD is in the extension of the predicate (to figure out whether Black Lightning fights crime, for instance), you need to know about comic books. In general, the extension of a predicate is the result of an interpretation *along with* some facts.

Sometimes it is possible to list all of the things that are in the extension of a predicate. Instead of writing a schematic English sentence, we can write down the extension as a set of things. Suppose we wanted to add a one-place predicate M to the key above. We want Mx to mean 'x lives in Wayne Manor', so we write the extension as a set of characters:

$$\text{extension}(M) = \{\text{Bruce Wayne, Alfred the butler, Dick Grayson}\}$$

You do not need to know anything about comic books to be able to determine that, on this interpretation, Mw is true: Bruce Wayne is just specified to be one of the things that is M. Similarly, $\exists xMx$ is obviously true on this interpretation: There is at least one member of the UD that is an M— in fact, there are three of them.

What about the sentence $\forall x Mx$? The sentence is false, because it is not true that all members of the UD are M. It requires the barest minimum of knowledge about comic books to know that there are other characters besides just these three. Although we specified the extension of M in a formally precise way, we still specified the UD with an English language description. Formally speaking, a UD is just a set of members.

The formal significance of a predicate is determined by its extension, but what should we say about constants like b and w? The meaning of a constant determines which member of the UD is picked out by the constant. The individual that the constant picks out is called the REFERENT of the constant. Both b and w have the same referent, since they both refer to the same comic book character. You can think of a constant letter as a name and the referent as the thing named. In English, we can use the different names 'Batman' and 'Bruce Wayne' to refer to the same comic book character. In this interpretation, we can use the different constants 'b' and 'w' to refer to the same member of the UD.

Sets

We use curly brackets '{' and '}' to denote sets. The members of the set can be listed in any order, separated by commas. The fact that sets can be in any order is important, because it means that {foo, bar} and {bar, foo} are the same set.

It is possible to have a set with no members in it. This is called the EMPTY SET. The empty set is sometimes written as {}, but usually it is written as the single symbol $\emptyset$.

Models

As we have seen, an interpretation in QL is only formally significant insofar as it determines a UD, an extension for each predicate, and a referent for each constant. We call this formal structure a MODEL for QL.

To see how this works, consider this symbolization key:

UD: People who played as part of the Three Stooges
Hx: x had head hair.
f: Mister Fine

If you do not know anything about the Three Stooges, you will not be able to say which sentences of QL are true on this interpretation. Perhaps you just remember Larry, Curly, and Moe. Is the sentence Hf true or false? It depends on which of the stooges is Mister Fine.

What is the model that corresponds to this interpretation? There were six people who played as part of the Three Stooges over the years, so the UD will have six members: Larry Fine, Moe Howard, Curly Howard, Shemp Howard, Joe Besser, and Curly Joe DeRita. Curly, Joe, and Curly Joe were the only completely bald stooges. The result is this model:

$$\begin{aligned} \text{UD} &= \{\text{Larry, Curly, Moe, Shemp, Joe, Curly Joe}\} \\ \text{extension}(H) &= \{\text{Larry, Moe, Shemp}\} \\ \text{referent}(f) &= \text{Larry} \end{aligned}$$

You do not need to know anything about the Three Stooges in order to evaluate whether sentences are true or false in this *model*. Hf is true, since the referent of f (Larry) is in the extension of H. Both $\exists xHx$ and $\exists x\neg Hx$ are true, since there is at least one member of the UD that is in the extension of H and at least one member that is not in the extension of H. In this way, the model captures all of the formal significance of the interpretation.

Now consider this interpretation:

UD: whole numbers less than 10
Ex: x is even.
Nx: x is negative.
Lxy: x is less than y.
Txyz: x times y equals z.

What is the model that goes with this interpretation? The UD is the set $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

The extension of a one-place predicate like E or N is just the subset of the UD of which the predicate is true. Roughly speaking, the extension of the predicate E is the set of Es in the UD. The extension of E is the subset $\{2, 4, 6, 8\}$. There are many even numbers besides these four, but these are the only members of the UD that are even. There are no negative numbers in the UD, so N has an empty extension; i.e. extension$(N) = \emptyset$.

The extension of a two-place predicate like L is somewhat vexing. It seems as if the extension of L ought to contain 1, since 1 is less than all the other numbers; it ought to contain 2, since 2 is less than all of the other numbers besides 1; and so on. Every member of the UD besides 9 is less than some member of the UD. What would happen if we just wrote extension$(L) = \{1, 2, 3, 4, 5, 6, 7, 8\}$?

The problem is that sets can be written in any order, so this would be the same as writing extension$(L) = \{8, 7, 6, 5, 4, 3, 2, 1\}$. This does not tell us which of the members of the set are less than which other members.

We need some way of showing that 1 is less than 8 but that 8 is not less than 1. The solution is to have the extension of L consist of pairs of numbers. An

ORDERED PAIR is like a set with two members, except that the order *does* matter. We write ordered pairs with angle brackets '<' and '>'. The ordered pair <foo, bar> is different than the ordered pair <bar, foo>. The extension of L is a collection of ordered pairs, all of the pairs of numbers in the UD such that the first number is less than the second. Writing this out completely:

> extension(L) = {<1,2>, <1,3>, <1,4>, <1,5>, <1,6>, <1,7>, <1,8>, <1,9>, <2,3>, <2,4>, <2,5>, <2,6>, <2,7>, <2,8>, <2,9>, <3,4>, <3,5>, <3,6>, <3,7>, <3,8>, <3,9>, <4,5>, <4,6>, <4,7>, <4,8>, <4,9>, <5,6>, <5,7>, <5,8>, <5,9>, <6,7>, <6,8>, <6,9>, <7,8>, <7,9>, <8,9>}

Three-place predicates will work similarly; the extension of a three-place predicate is a set of ordered triples where the predicate is true of those three things *in that order*. So the extension of T in this model will contain ordered triples like <2,4,8>, because $2 \times 4 = 8$.

Generally, the extension of an n-place predicate is a set of all ordered n-tuples $\langle a_1, a_2, \ldots, a_n \rangle$ such that a_1–a_n are members of the UD and the predicate is true of a_1–a_n in that order.

6.3 Semantics for identity

Identity is a special predicate of QL. We write it a bit differently than other two-place predicates: $x = y$ instead of Ixy. We also do not need to include it in a symbolization key. The sentence $x = y$ always means 'x is identical to y,' and it cannot be interpreted to mean anything else. In the same way, when you construct a model, you do not get to pick and choose which ordered pairs go into the extension of the identity predicate. It always contains just the ordered pair of each object in the UD with itself.

The sentence $\forall x Ixx$, which contains an ordinary two-place predicate, is contingent. Whether it is true for an interpretation depends on how you interpret I, and whether it is true in a model depends on the extension of I.

The sentence $\forall x\ x = x$ is a tautology. The extension of identity will always make it true.

Notice that although identity always has the same interpretation, it does not always have the same extension. The extension of identity depends on the UD. If the UD in a model is the set {Doug}, then extension(=) in that model is {<Doug, Doug>}. If the UD is the set {Doug, Omar}, then extension(=) in that model is {<Doug, Doug>, <Omar, Omar>}. And so on.

If the referent of two constants is the same, then anything which is true of one is true of the other. For example, if referent(a) = referent(b), then $Aa \leftrightarrow Ab$,

$Ba \leftrightarrow Bb$, $Ca \leftrightarrow Cb$, $Rca \leftrightarrow Rcb$, $\forall xRxa \leftrightarrow \forall xRxb$, and so on for any two sentences containing a and b. However, the reverse is not true.

It is possible that anything which is true of a is also true of b, yet for a and b still to have different referents. This may seem puzzling, but it is easy to construct a model that shows this. Consider this model:

$$\begin{aligned} \text{UD} &= \{\text{Rosencrantz, Guildenstern}\} \\ \text{referent}(a) &= \text{Rosencrantz} \\ \text{referent}(b) &= \text{Guildenstern} \\ \text{for all predicates } \mathcal{P}\text{, extension}(\mathcal{P}) &= \emptyset \\ \text{extension}(=) &= \{<\text{Rosencrantz, Rosencrantz}>, \\ &\quad\ <\text{Guildenstern, Guildenstern}>\} \end{aligned}$$

This specifies an extension for every predicate of QL: All the infinitely-many predicates are empty. This means that both Aa and Ab are false, and they are equivalent; both Ba and Bb are false; and so on for any two sentences that contain a and b. Yet a and b refer to different things. We have written out the extension of identity to make this clear: The ordered pair $\langle \text{referent}(a), \text{referent}(b) \rangle$ is not in it. In this model, $a = b$ is false and $a \neq b$ is true.

6.4 Working with models

We will use the double turnstile symbol for QL much as we did for SL. '$\mathcal{A} \models \mathcal{B}$' means that '$\mathcal{A}$ entails $\mathcal{B}$': When $\mathcal{A}$ and $\mathcal{B}$ are two sentences of QL, $\mathcal{A} \models \mathcal{B}$ means that there is no model in which $\mathcal{A}$ is true and $\mathcal{B}$ is false. $\models \mathcal{A}$ means that $\mathcal{A}$ is true in every model.

This allows us to give definitions for various concepts in QL. Because we are using the same symbol, these definitions will look similar to the definitions in SL. Remember, however, that the definitions in QL are in terms of *models* rather than in terms of truth value assignments.

> A TAUTOLOGY IN QL is a sentence $\mathcal{A}$ that is true in every model; i.e., $\models \mathcal{A}$.
>
> A CONTRADICTION IN QL is a sentence $\mathcal{A}$ that is false in every model; i.e., $\models \neg\mathcal{A}$.
>
> A sentence is CONTINGENT IN QL if and only if it is neither a tautology nor a contradiction.
>
> An argument " $\mathcal{P}_1, \mathcal{P}_2, \cdots, \therefore\ \mathcal{C}$ " is VALID IN QL if and only if there is no model in which all of the premises are true and the conclusion is false; i.e., $\{\mathcal{P}_1, \mathcal{P}_2, \cdots\} \models \mathcal{C}$. It is INVALID IN QL otherwise.
>
> Two sentences $\mathcal{A}$ and $\mathcal{B}$ are LOGICALLY EQUIVALENT IN QL if and only if both $\mathcal{A} \models \mathcal{B}$ and $\mathcal{B} \models \mathcal{A}$.

The set $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \cdots\}$ is CONSISTENT IN QL if and only if there is at least one model in which all of the sentences are true. The set is INCONSISTENT IN QL if and if only there is no such model.

Constructing models

Suppose we want to show that $\forall x Axx \rightarrow Bd$ is *not* a tautology. This requires showing that the sentence is not true in every model; i.e., that it is false in some model. If we can provide just one model in which the sentence is false, then we will have shown that the sentence is not a tautology.

What would such a model look like? In order for $\forall x Axx \rightarrow Bd$ to be false, the antecedent ($\forall x Axx$) must be true, and the consequent (Bd) must be false.

To construct such a model, we start with a UD. It will be easier to specify extensions for predicates if we have a small UD, so start with a UD that has just one member. Formally, this single member might be anything. Let's say it is the city of Paris.

We want $\forall x Axx$ to be true, so we want all members of the UD to be paired with themself in the extension of A; this means that the extension of A must be {<Paris,Paris>}.

We want Bd to be false, so the referent of d must not be in the extension of B. We give B an empty extension.

Since Paris is the only member of the UD, it must be the referent of d. The model we have constructed looks like this:

$$\begin{aligned} \text{UD} &= \{\text{Paris}\} \\ \text{extension}(A) &= \{<\text{Paris,Paris}>\} \\ \text{extension}(B) &= \emptyset \\ \text{referent}(d) &= \text{Paris} \end{aligned}$$

Strictly speaking, a model specifies an extension for *every* predicate of QL and a referent for *every* constant. As such, it is generally impossible to write down a complete model. That would require writing down infinitely many extensions and infinitely many referents. However, we do not need to consider every predicate in order to show that there are models in which $\forall x Axx \rightarrow Bd$ is false. Predicates like H and constants like f_{13} make no difference to the truth or falsity of this sentence. It is enough to specify extensions for A and B and a referent for d, as we have done. This provides a *partial model* in which the sentence is false.

Perhaps you are wondering: What does the predicate A mean in English? The partial model could correspond to an interpretation like this one:

UD: Paris

Axy**:** x is in the same country as y.
Bx**:** x was founded in the 20th century.
d**:** the City of Lights

However, all that the partial model tells us is that A is a predicate which is true of Paris and Paris. There are indefinitely many predicates in English that have this extension. Axy might instead translate 'x is the same size as y' or 'x and y are both cities.' Similarly, Bx is some predicate that does not apply to Paris; it might instead translate 'x is on an island' or 'x is a subcompact car.' When we specify the extensions of A and B, we do not specify what English predicates A and B should be used to translate. We are concerned with whether the $\forall xAxx \rightarrow Bd$ comes out true or false, and all that matters for truth and falsity in QL is the information in the model: the UD, the extensions of predicates, and the referents of constants.

We can just as easily show that $\forall xAxx \rightarrow Bd$ is not a contradiction. We need only specify a model in which $\forall xAxx \rightarrow Bd$ is true; i.e., a model in which either $\forall xAxx$ is false or Bd is true. Here is one such partial model:

$$\begin{aligned} \text{UD} &= \{\text{Paris}\} \\ \text{extension}(A) &= \{<\text{Paris,Paris}>\} \\ \text{extension}(B) &= \{\text{Paris}\} \\ \text{referent}(d) &= \text{Paris} \end{aligned}$$

We have now shown that $\forall xAxx \rightarrow Bd$ is neither a tautology nor a contradiction. By the definition of 'contingent in QL,' this means that $\forall xAxx \rightarrow Bd$ is contingent. In general, showing that a sentence is contingent will require two models: one in which the sentence is true and another in which the sentence is false.

Suppose we want to show that $\forall xSx$ and $\exists xSx$ are not logically equivalent. We need to construct a model in which the two sentences have different truth values; we want one of them to be true and the other to be false. We start by specifying a UD. Again, we make the UD small so that we can specify extensions easily. We will need at least two members. Let the UD be {Duke, Miles}. (If we chose a UD with only one member, the two sentences would end up with the same truth value. In order to see why, try constructing some partial models with one-member UDs.)

We can make $\exists xSx$ true by including something in the extension of S, and we can make $\forall xSx$ false by leaving something out of the extension of S. It does not matter which one we include and which one we leave out. Making Duke the only S, we get a partial model that looks like this:

$$\begin{aligned} \text{UD} &= \{\text{Duke, Miles}\} \\ \text{extension}(S) &= \{\text{Duke}\} \end{aligned}$$

This partial model shows that the two sentences are *not* logically equivalent.

Back on p. 80, we said that this argument would be invalid in QL:

$(Rc \wedge K_1c) \wedge Tc$
$\therefore\ Tc \wedge K_2c$

In order to show that it is invalid, we need to show that there is some model in which the premises are true and the conclusion is false. We can construct such a model deliberately. Here is one way to do it:

$$\begin{aligned}
\text{UD} &= \{\text{Björk}\}\\
\text{extension}(T) &= \{\text{Björk}\}\\
\text{extension}(K_1) &= \{\text{Björk}\}\\
\text{extension}(K_2) &= \emptyset\\
\text{extension}(R) &= \{\text{Björk}\}\\
\text{referent}(c) &= \text{Björk}
\end{aligned}$$

Similarly, we can show that a set of sentences is consistent by constructing a model in which all of the sentences are true.

Reasoning about all models

We can show that a sentence is *not* a tautology just by providing one carefully specified model: a model in which the sentence is false. To show that something is a tautology, on the other hand, it would not be enough to construct ten, one hundred, or even a thousand models in which the sentence is true. It is only a tautology if it is true in *every* model, and there are infinitely many models. This cannot be avoided just by constructing partial models, because there are infinitely many partial models.

Consider, for example, the sentence $Raa \leftrightarrow Raa$. There are two logically distinct partial models of this sentence that have a 1-member UD. There are 32 distinct partial models that have a 2-member UD. There are 1526 distinct partial models that have a 3-member UD. There are 262,144 distinct partial models that have a 4-member UD. And so on to infinity. In order to show that this sentence is a tautology, we need to show something about all of these models. There is no hope of doing so by dealing with them one at a time.

Nevertheless, $Raa \leftrightarrow Raa$ is obviously a tautology. We can prove it with a simple argument:

> There are two kinds of models: those in which $\langle\text{referent}(a), \text{referent}(a)\rangle$ is in the extension of R and those in which it is not. In the first kind of model, Raa is true; by the truth table for the biconditional, $Raa \leftrightarrow Raa$ is also true. In the second kind of model, Raa is false; this makes $Raa \leftrightarrow Raa$ true. Since the sentence is true in both kinds of model, and since every model is one of the two kinds, $Raa \leftrightarrow Raa$ is true in every model. Therefore, it is a tautology.

This argument is valid, of course, and its conclusion is true. However, it is not an argument in QL. Rather, it is an argument in English *about* QL; it is an argument in the metalanguage. There is no formal procedure for evaluating or constructing natural language arguments like this one. The imprecision of natural language is the very reason we began thinking about formal languages.

There are further difficulties with this approach.

Consider the sentence $\forall x(Rxx \rightarrow Rxx)$, another obvious tautology. It might be tempting to reason in this way: '$Rxx \rightarrow Rxx$ is true in every model, so $\forall x(Rxx \rightarrow Rxx)$ must be true.' The problem is that $Rxx \rightarrow Rxx$ is *not* true in every model. It is not a sentence, and so it is *neither* true *nor* false. We do not yet have the vocabulary to say what we want to say about $Rxx \rightarrow Rxx$. In the next section, we introduce the concept of *satisfaction*; after doing so, we will be better able to provide an argument that $\forall x(Rxx \rightarrow Rxx)$ is a tautology.

It is necessary to reason about an infinity of models to show that a sentence is a tautology. Similarly, it is necessary to reason about an infinity of models to show that a sentence is a contradition, that two sentences are equivalent, that a set of sentences is inconsistent, or that an argument is valid. There are other things we can show by carefully constructing a model or two. Table 6.1 summarizes which things are which.

Table 6.1: It is relatively easy to answer a question if you can do it by constructing a model or two. It is much harder if you need to reason about all possible models. This table shows when constructing models is enough.

	YES	NO
Is $\mathcal{A}$ a tautology?	show that $\mathcal{A}$ must be true in any model	*construct a model* in which $\mathcal{A}$ is false
Is $\mathcal{A}$ a contradiction?	show that $\mathcal{A}$ must be false in any model	*construct a model* in which $\mathcal{A}$ is true
Is $\mathcal{A}$ contingent?	*construct two models*, one in which $\mathcal{A}$ is true and another in which $\mathcal{A}$ is false	either show that $\mathcal{A}$ is a tautology or show that $\mathcal{A}$ is a contradiction
Are $\mathcal{A}$ and $\mathcal{B}$ equivalent?	show that $\mathcal{A}$ and $\mathcal{B}$ must have the same truth value in any model	*construct a model* in which $\mathcal{A}$ and $\mathcal{B}$ have different truth values
Is the set $\mathbb{A}$ consistent?	*construct a model* in which all the sentences in $\mathbb{A}$ are true	show that the sentences could not all be true in any model
Is the argument '$\mathcal{P}$, $\therefore$ $\mathcal{C}$' valid?	show that any model in which $\mathcal{P}$ is true must be a model in which $\mathcal{C}$ is true	*construct a model* in which $\mathcal{P}$ is true and $\mathcal{C}$ is false

6.5 Truth in QL

For SL, we split the definition of truth into two parts: a truth value assignment (a) for sentence letters and a truth function (v) for all sentences. The truth function covered the way that complex sentences could be built out of sentence letters and connectives.

In the same way that truth for SL is always *truth given a truth value assignment*, truth for QL is *truth in a model*. The simplest atomic sentence of QL consists of a one-place predicate followed by a constant, like Pj. It is true in a model $\mathbb{M}$ if and only if the referent of j is in the extension of P in $\mathbb{M}$.

We could go on in this way to define truth for all atomic sentences that contain only predicates and constants: Consider any sentence of the form $\mathcal{R}c_1 \ldots c_n$ where $\mathcal{R}$ is an n-place predicate and the cs are constants. It is true in $\mathbb{M}$ if and only if $\langle \text{referent}(c_1), \ldots, \text{referent}(c_n)\rangle$ is in extension($\mathcal{R}$) in $\mathbb{M}$.

We could then define truth for sentences built up with sentential connectives in the same way we did for SL. For example, the sentence $(Pj \rightarrow Mda)$ is true in $\mathbb{M}$ if either Pj is false in $\mathbb{M}$ or Mda is true in $\mathbb{M}$.

Unfortunately, this approach will fail when we consider sentences containing quantifiers. Consider $\forall x Px$. When is it true in a model $\mathbb{M}$? The answer cannot depend on whether Px is true or false in $\mathbb{M}$, because the x in Px is a free variable. Px is not a sentence. It is neither true nor false.

We were able to give a recursive definition of truth for SL because every well-formed formula of SL has a truth value. This is not true in QL, so we cannot define truth by starting with the truth of atomic sentences and building up. We also need to consider the atomic formulae which are not sentences. In order to do this we will define *satisfaction*; every well-formed formula of QL will be satisfied or not satisfied, even if it does not have a truth value. We will then be able to define *truth* for sentences of QL in terms of satisfaction.

Practice Exercises

$\star$ **Part A** Determine whether each sentence is true or false in the model given.

$$
\begin{aligned}
\text{UD} &= \{\text{Corwin, Benedict}\}\\
\text{extension}(A) &= \{\text{Corwin, Benedict}\}\\
\text{extension}(B) &= \{\text{Benedict}\}\\
\text{extension}(N) &= \emptyset\\
\text{referent}(c) &= \text{Corwin}
\end{aligned}
$$

1. Bc

2. $Ac \leftrightarrow \neg Nc$
3. $Nc \rightarrow (Ac \vee Bc)$
4. $\forall x Ax$
5. $\forall x \neg Bx$
6. $\exists x(Ax \wedge Bx)$
7. $\exists x(Ax \rightarrow Nx)$
8. $\forall x(Nx \vee \neg Nx)$
9. $\exists x Bx \rightarrow \forall x Ax$

⋆ **Part B** Determine whether each sentence is true or false in the model given.

UD = {Waylan, Willy, Johnny}
extension(H) = {Waylan, Willy, Johnny}
extension(W) = {Waylan, Willy}
extension(R) = {<Waylan, Willy>,<Willy, Johnny>,<Johnny, Waylan>}
referent(m) = Johnny

1. $\exists x(Rxm \wedge Rmx)$
2. $\forall x(Rxm \vee Rmx)$
3. $\forall x(Hx \leftrightarrow Wx)$
4. $\forall x(Rxm \rightarrow Wx)$
5. $\forall x\big[Wx \rightarrow (Hx \wedge Wx)\big]$
6. $\exists x Rxx$
7. $\exists x \exists y Rxy$
8. $\forall x \forall y Rxy$
9. $\forall x \forall y(Rxy \vee Ryx)$
10. $\forall x \forall y \forall z\big[(Rxy \wedge Ryz) \rightarrow Rxz\big]$

Part C Determine whether each sentence is true or false in the model given.

UD = {Lemmy, Courtney, Eddy}
extension(G) = {Lemmy, Courtney, Eddy}
extension(H) = {Courtney}
extension(M) = {Lemmy, Eddy}
referent(c) = Courtney
referent(e) = Eddy

1. Hc
2. He
3. $Mc \vee Me$
4. $Gc \vee \neg Gc$
5. $Mc \rightarrow Gc$
6. $\exists x Hx$
7. $\forall x Hx$
8. $\exists x \neg Mx$
9. $\exists x(Hx \wedge Gx)$

10. $\exists x(Mx \wedge Gx)$
11. $\forall x(Hx \vee Mx)$
12. $\exists xHx \wedge \exists xMx$
13. $\forall x(Hx \leftrightarrow \neg Mx)$
14. $\exists xGx \wedge \exists x\neg Gx$
15. $\forall x\exists y(Gx \wedge Hy)$

⋆ **Part D** Write out the model that corresponds to the interpretation given.

UD: natural numbers from 10 to 13
Ox: x is odd.
Sx: x is less than 7.
Tx: x is a two-digit number.
Ux: x is thought to be unlucky.
Nxy: x is the next number after y.

Part E Show that each of the following is contingent.

⋆ 1. $Da \wedge Db$
⋆ 2. $\exists xTxh$
⋆ 3. $Pm \wedge \neg\forall xPx$
4. $\forall zJz \leftrightarrow \exists yJy$
5. $\forall x(Wxmn \vee \exists yLxy)$
6. $\exists x(Gx \rightarrow \forall yMy)$

⋆ **Part F** Show that the following pairs of sentences are not logically equivalent.

1. Ja, Ka
2. $\exists xJx$, Jm
3. $\forall xRxx$, $\exists xRxx$
4. $\exists xPx \rightarrow Qc$, $\exists x(Px \rightarrow Qc)$
5. $\forall x(Px \rightarrow \neg Qx)$, $\exists x(Px \wedge \neg Qx)$
6. $\exists x(Px \wedge Qx)$, $\exists x(Px \rightarrow Qx)$
7. $\forall x(Px \rightarrow Qx)$, $\forall x(Px \wedge Qx)$
8. $\forall x\exists yRxy$, $\exists x\forall yRxy$
9. $\forall x\exists yRxy$, $\forall x\exists yRyx$

Part G Show that the following sets of sentences are consistent.

1. {Ma, ¬Na, Pa, ¬Qa}
2. $\{Lee, Lef, \neg Lfe, \neg Lff\}$
3. $\{\neg(Ma \wedge \exists xAx), Ma \vee Fa, \forall x(Fx \rightarrow Ax)\}$
4. $\{Ma \vee Mb, Ma \rightarrow \forall x\neg Mx\}$
5. $\{\forall yGy, \forall x(Gx \rightarrow Hx), \exists y\neg Iy\}$
6. $\{\exists x(Bx \vee Ax), \forall x\neg Cx, \forall x\big[(Ax \wedge Bx) \rightarrow Cx\big]\}$

7. $\{\exists x Xx,\ \exists x Yx,\ \forall x(Xx \leftrightarrow \neg Yx)\}$
8. $\{\forall x(Px \vee Qx),\ \exists x \neg(Qx \wedge Px)\}$
9. $\{\exists z(Nz \wedge Ozz),\ \forall x \forall y(Oxy \rightarrow Oyx)\}$
10. $\{\neg \exists x \forall y Rxy,\ \forall x \exists y Rxy\}$

Part H Construct models to show that the following arguments are invalid.

1. $\forall x(Ax \rightarrow Bx)$, $\therefore \exists x Bx$
2. $\forall x(Rx \rightarrow Dx)$, $\forall x(Rx \rightarrow Fx)$, $\therefore \exists x(Dx \wedge Fx)$
3. $\exists x(Px \rightarrow Qx)$, $\therefore \exists x Px$
4. $Na \wedge Nb \wedge Nc$, $\therefore \forall x Nx$
5. Rde, $\exists x Rxd$, $\therefore Red$
6. $\exists x(Ex \wedge Fx)$, $\exists x Fx \rightarrow \exists x Gx$, $\therefore \exists x(Ex \wedge Gx)$
7. $\forall x Oxc$, $\forall x Ocx$, $\therefore \forall x Oxx$
8. $\exists x(Jx \wedge Kx)$, $\exists x \neg Kx$, $\exists x \neg Jx$, $\therefore \exists x(\neg Jx \wedge \neg Kx)$
9. $Lab \rightarrow \forall x Lxb$, $\exists x Lxb$, $\therefore Lbb$

Part I

⋆ 1. Show that $\{\neg Raa, \forall x(x = a \vee Rxa)\}$ is consistent.
⋆ 2. Show that $\{\forall x \forall y \forall z(x = y \vee y = z \vee x = z), \exists x \exists y\ x \neq y\}$ is consistent.
⋆ 3. Show that $\{\forall x \forall y\ x = y, \exists x\ x \neq a\}$ is inconsistent.
4. Show that $\exists x(x = h \wedge x = i)$ is contingent.
5. Show that $\{\exists x \exists y(Zx \wedge Zy \wedge x = y),\ \neg Zd,\ d = s\}$ is consistent.
6. Show that '$\forall x(Dx \rightarrow \exists y Tyx)$ $\therefore \exists y \exists z\ y \neq z$' is invalid.

Chapter 7

Proofs in QL

7.1 Rules for quantifiers

For proofs in QL, we use all of the basic rules of SL plus four new basic rules: both introduction and elimination rules for each of the quantifiers.

Since all of the derived rules of SL are derived from the basic rules, they will also hold in QL. We will add another derived rule, a replacement rule called quantifier negation.

Substitution instances

In order to concisely state the rules for the quantifiers, we need a way to mark the relation between quantified sentences and their instances. For example, the sentence Pa is a particular instance of the general claim $\forall x Px$.

For a wff $\mathcal{A}$, a constant c, and a variable χ, define a SUBSTITUTION INSTANCE of $\forall\chi\mathcal{A}$ or $\exists\chi\mathcal{A}$ is the wff that we get by replacing every occurrence of χ in $\mathcal{A}$ with c. We call c the INSTANTIATING CONSTANT.

To underscore the fact that the variable χ is replaced by the instantiating constant c, we will write the original quantified expressions as $\forall\chi\mathcal{A}\chi$ and $\exists\chi\mathcal{A}\chi$. And we will write the substitution instance $\mathcal{A}c$.

Note that $\mathcal{A}$, χ, and c are all meta-variables. That is, they are stand-ins for any wff, variable, and constant whatsoever. And when we write $\mathcal{A}c$, the constant c may occur multiple times in the wff $\mathcal{A}$.

For example:

- $Aa \to Ba$, $Af \to Bf$, and $Ak \to Bk$ are all substitution instances of $\forall x(Ax \to Bx)$; the instantiating constants are a, f, and k, respectively.

- Raj, Rdj, and Rjj are substitution instances of $\exists zRzj$; the instantiating constants are a, d, and j, respectively.

Universal elimination

If you have $\forall xAx$, it is legitimate to infer that anything is an A. You can infer Aa, Ab, Az, Ad_3. You can infer any substitution instance, $A\mathcal{c}$ for any constant $\mathcal{c}$.

This is the general form of the universal elimination rule ($\forall$E):

m	$\forall \mathcal{X}\mathcal{A}\mathcal{X}$	
	$\mathcal{A}\mathcal{c}$	$\forall$E m

When using the $\forall$E rule, you write the substituted sentence with the constant $\mathcal{c}$ replacing all occurrences of the variable $\mathcal{X}$ in $\mathcal{A}$. For example:

1	$\forall x(Mx \to Rxd)$	
2	$Ma \to Rad$	$\forall$E 1
3	$Md \to Rdd$	$\forall$E 1

Existential introduction

It is legitimate to infer $\exists xPx$ if you know that *something* is a P. It might be any particular thing at all. For example, if you have Pa available in the proof, then $\exists xPx$ follows.

This is the existential introduction rule ($\exists$I):

m	$\mathcal{A}\mathcal{c}$	
	$\exists \mathcal{X}\mathcal{A}\mathcal{X}$	$\exists$I m

It is important to notice that the variable $\mathcal{X}$ does not need to replace all occurrences of the constant $\mathcal{c}$. You can decide which occurrences to replace and which to leave in place. For example:

1	$Ma \to Rad$	
2	$\exists x(Ma \to Rax)$	$\exists$I 1
3	$\exists x(Mx \to Rxd)$	$\exists$I 1
4	$\exists x(Mx \to Rad)$	$\exists$I 1
5	$\exists y \exists x(Mx \to Ryd)$	$\exists$I 4
6	$\exists z \exists y \exists x(Mx \to Ryz)$	$\exists$I 5

Universal introduction

A universal claim like $\forall x Px$ would be proven if every substitution instance of it had been proven. That is, if every sentence Pa, Pb, ... were available in a proof, then you would certainly be entitled to claim $\forall x Px$. Alas, there is no hope of proving *every* substitution instance. That would require proving Pa, Pb, ..., Pj_2, ..., Ps_7, ..., and so on to infinity. There are infinitely many constants in QL, and so this process would never come to an end.

Consider instead a simple argument: $\forall x Mx$, $\therefore \forall y My$

It makes no difference to the meaning of the sentence whether we use the variable x or the variable y, so this argument is obviously valid. Suppose we begin in this way:

1	$\forall x Mx$	want $\forall y My$
2	Ma	$\forall$E 1

We have derived Ma. Nothing stops us from using the same justification to derive Mb, ..., Mj_2, ..., Ms_7, ..., and so on until we run out of space or patience. We have effectively shown the way to prove $M\mathcal{c}$ for any constant $\mathcal{c}$. From this, $\forall y My$ follows.

1	$\forall x Mx$	
2	Ma	$\forall$E 1
3	$\forall y My$	$\forall$I 2

It is important here that a was just some arbitrary constant. We had not made any special assumptions about it. If Ma were a premise of the argument, then this would not show anything about *all* y. For example:

1	$\forall x Rxa$	
2	Raa	$\forall$E 1
3	$\forall y Ryy$	not allowed!

This is the schematic form of the universal introduction rule ($\forall$I):

m	$\mathcal{A}c^*$	
	$\forall \chi \mathcal{A}\chi$	$\forall$I m

* The constant c must not occur in any undischarged assumption.

Note that we can do this for any constant that does not occur in an undischarged assumption and for any variable.

Note also that the constant may not occur in any *undischarged* assumption, but it may occur as the assumption of a subproof that we have already closed. For example, we can prove $\forall z(Dz \to Dz)$ without any premises.

1	$\quad Df$	want Df
2	$\quad Df$	R 1
3	$Df \to Df$	$\to$I 1–2
4	$\forall z(Dz \to Dz)$	$\forall$I 3

Existential elimination

A sentence with an existential quantifier tells us that there is *some* member of the UD that satisfies a formula. For example, $\exists xSx$ tells us (roughly) that there is at least one S. It does not tell us *which* member of the UD satisfies S, however. We cannot immediately conclude Sa, Sf_{23}, or any other substitution instance of the sentence. What can we do?

Suppose that we knew both $\exists xSx$ and $\forall x(Sx \to Tx)$. We could reason in this way:

> Since $\exists xSx$, there is something that is an S. We do not know which constants refer to this thing, if any do, so call this thing 'Ishmael'. From $\forall x(Sx \to Tx)$, it follows that if Ishmael is an S, then it is a T. Therefore, Ishmael is a T. Because Ishmael is a T, we know that $\exists xTx$.

In this paragraph, we introduced a name for the thing that is an S. We gave it an arbitrary name ('Ishmael') so that we could reason about it and derive some consequences from there being an S. Since ‘Ishmael’ is just a bogus name introduced for the purpose of the proof and not a genuine constant, we could not mention it in the conclusion. Yet we could derive a sentence that does not mention Ishmael; namely, $\exists x Tx$. This sentence does follow from the two premises.

We want the existential elimination rule to work in a similar way. Yet since English language words like ‘Ishmael’ are not symbols of QL, we cannot use them in formal proofs. Instead, we will use constants of QL which do not otherwise appear in the proof.

A constant that is used to stand in for whatever it is that satisfies an existential claim is called a PROXY. Reasoning with the proxy must all occur inside a subproof, and the proxy cannot be a constant that is doing work elsewhere in the proof.

This is the schematic form of the existential elimination rule (∃E):

m	$\exists \chi \mathcal{A}\chi$	
n	$\quad \mathcal{A}c^*$	
p	$\quad \mathcal{B}$	
	$\mathcal{B}$	∃E m, n–p

* The constant c must not appear in $\exists \chi \mathcal{A}\chi$, in $\mathcal{B}$, or in any undischarged assumption.

Since the proxy constant is just a place holder that we use inside the subproof, it cannot be something that we know anything particular about. So it cannot appear in the original sentence $\exists \chi \mathcal{A}\chi$ or in an undischarged assumption. Moreover, we do not learn anything about the proxy constant by using the ∃E rule. So it cannot appear in $\mathcal{B}$, the sentence you prove using ∃E.

The easiest way to satisfy these requirements is to pick an entirely new constant when you start the subproof, and then not to use that constant anywhere else in the proof. Once you close the subproof, do not mention it again.

With this rule, we can give a formal proof that $\exists x Sx$ and $\forall x(Sx \rightarrow Tx)$ together entail $\exists x Tx$.

1	$\exists x Sx$	
2	$\forall x(Sx \to Tx)$	want $\exists x Tx$
3	$\quad Si$	
4	$\quad Si \to Ti$	$\forall$E 2
5	$\quad Ti$	$\to$E 3, 4
6	$\quad \exists x Tx$	$\exists$I 5
7	$\exists x Tx$	$\exists$E 1, 3–6

Notice that this has effectively the same structure as the English-language argument with which we began, except that the subproof uses the proxy constant 'i' rather than the bogus name 'Ishmael'.

Quantifier negation

When translating from English to QL, we noted that $\neg\exists x\neg\mathcal{A}$ is logically equivalent to $\forall x\mathcal{A}$. In QL, they are provably equivalent. We can prove one half of the equivalence with a rather gruesome proof:

1	$\forall x Ax$	want $\neg\exists x\neg Ax$
2	$\quad \exists x\neg Ax$	for reductio
3	$\quad\quad \neg Ac$	for $\exists$E
4	$\quad\quad\quad \forall x Ax$	for reductio
5	$\quad\quad\quad Ac$	$\forall$E 1
6	$\quad\quad\quad \neg Ac$	R 3
7	$\quad\quad \neg\forall x Ax$	$\neg$I 4–6
8	$\quad \forall x Ax$	R 1
9	$\quad \neg\forall x Ax$	$\exists$E 2, 3–7
10	$\neg\exists x\neg Ax$	$\neg$I 2–9

In order to show that the two sentences are genuinely equivalent, we need a second proof that assumes $\neg\exists x\neg\mathcal{A}$ and derives $\forall x\mathcal{A}$. We leave that proof as an exercise for the reader.

It will often be useful to translate between quantifiers by adding or subtracting negations in this way, so we add two derived rules for this purpose. These rules are called quantifier negation (QN):

$$\neg\forall\chi\mathcal{A} \Longleftrightarrow \exists\chi\neg\mathcal{A}$$
$$\neg\exists\chi\mathcal{A} \Longleftrightarrow \forall\chi\neg\mathcal{A} \qquad \text{QN}$$

Since QN is a replacement rule, it can be used on whole sentences or on subformulae.

7.2 Rules for identity

The identity predicate is not part of QL, but we add it when we need to symbolize certain sentences. For proofs involving identity, we add two rules of proof.

Suppose you know that many things that are true of a are also true of b. For example: $Aa \wedge Ab$, $Ba \wedge Bb$, $\neg Ca \wedge \neg Cb$, $Da \wedge Db$, $\neg Ea \wedge \neg Eb$, and so on. This would not be enough to justify the conclusion $a = b$. (See p. 106.) In general, there are no sentences that do not already contain the identity predicate that could justify the conclusion $a = b$. This means that the identity introduction rule will not justify $a = b$ or any other identity claim containing two different constants.

However, it is always true that $a = a$. In general, no premises are required in order to conclude that something is identical to itself. So this will be the identity introduction rule, abbreviated =I:

$$\big|\ c = c \qquad =\text{I}$$

Notice that the =I rule does not require referring to any prior lines of the proof. For any constant c, you can write $c = c$ on any point with only the =I rule as justification.

If you have shown that $a = b$, then anything that is true of a must also be true of b. For any sentence with a in it, you can replace some or all of the occurrences of a with b and produce an equivalent sentence. For example, if you already know Raa, then you are justified in concluding Rab, Rba, Rbb.

The identity elimination rule (=E) allows us to do this. It justifies replacing terms with other terms that are identical to it.

For writing the rule, we will introduce a new bit of symoblism. For a sentence $\mathcal{A}$ and constants c and d, $\mathcal{A}c\circlearrowleft d$ is a sentence produced by replacing some or all instances of c in $\mathcal{A}$ with d or replacing instances of d with c. This is not the same as a substitution instance, because one constant need not replace every occurrence of the other (although it may).

We can now concisely write =E in this way:

m	$\mathcal{c} = \mathcal{d}$	
n	$\mathcal{A}$	
	$\mathcal{A}\mathcal{c} \circlearrowleft \mathcal{d}$	=E m, n

To see the rules in action, consider this proof:

1	$\forall x \forall y\ x = y$	
2	$\exists x Bx$	
3	$\forall x(Bx \rightarrow \neg Cx)$	want $\neg\exists x Cx$
4	$\quad Be$	
5	$\quad \forall y\ e = y$	$\forall$E 1
6	$\quad e = f$	$\forall$E 5
7	$\quad Bf$	=E 6, 4
8	$\quad Bf \rightarrow \neg Cf$	$\forall$E 3
9	$\quad \neg Cf$	$\rightarrow$E 8, 7
10	$\neg Cf$	$\exists$E 2, 4–9
11	$\forall x \neg Cx$	$\forall$I 10
12	$\neg\exists x Cx$	QN 11

7.3 Proof strategy

There is no simple recipe for proofs, and there is no substitute for practice. Here, though, are some rules of thumb and strategies to keep in mind.

Work backwards from what you want. The ultimate goal is to derive the conclusion. Look at the conclusion and ask what the introduction rule is for its main logical operator. This gives you an idea of what should happen *just before* the last line of the proof. Then you can treat this line as if it were your goal. Ask what you could do to derive this new goal.

For example: If your conclusion is a conditional $\mathcal{A} \rightarrow \mathcal{B}$, plan to use the $\rightarrow$I rule. This requires starting a subproof in which you assume $\mathcal{A}$. In the subproof, you want to derive $\mathcal{B}$.

Work forwards from what you have. When you are starting a proof, look at the premises; later, look at the sentences that you have derived so far. Think about the elimination rules for the main operators of these sentences. These will tell you what your options are.

For example: If you have $\forall x\mathcal{A}$, think about instantiating it for any constant that might be helpful. If you have $\exists x\mathcal{A}$ and intend to use the $\exists$E rule, then you should assume $\mathcal{A}[c|x]$ for some c that is not in use and then derive a conclusion that does not contain c.

For a short proof, you might be able to eliminate the premises and introduce the conclusion. A long proof is formally just a number of short proofs linked together, so you can fill the gap by alternately working back from the conclusion and forward from the premises.

Change what you are looking at. Replacement rules can often make your life easier. If a proof seems impossible, try out some different substitutions.

For example: It is often difficult to prove a disjunction using the basic rules. If you want to show $\mathcal{A} \vee \mathcal{B}$, it is often easier to show $\neg\mathcal{A} \rightarrow \mathcal{B}$ and use the MC rule.

Showing $\neg\exists x\mathcal{A}$ can also be hard, and it is often easier to show $\forall x\neg\mathcal{A}$ and use the QN rule.

Some replacement rules should become second nature. If you see a negated disjunction, for instance, you should immediately think of DeMorgan's rule.

Do not forget indirect proof. If you cannot find a way to show something directly, try assuming its negation.

Remember that most proofs can be done either indirectly or directly. One way might be easier— or perhaps one sparks your imagination more than the other— but either one is formally legitimate.

Repeat as necessary. Once you have decided how you might be able to get to the conclusion, ask what you might be able to do with the premises. Then consider the target sentences again and ask how you might reach them.

Persist. Try different things. If one approach fails, then try something else.

7.4 Proofs and models

As you might already suspect, there is a connection between *theorems* and *tautologies*.

There is a formal way of showing that a sentence is a theorem: Prove it. For each line, we can check to see if that line follows by the cited rule. It may be hard to produce a twenty line proof, but it is not so hard to check each line of the proof and confirm that it is legitimate— and if each line of the proof individually is legitimate, then the whole proof is legitimate. Showing that a sentence is a tautology, though, requires reasoning in English about all possible models. There is no formal way of checking to see if the reasoning is sound. Given a choice between showing that a sentence is a theorem and showing that it is a tautology, it would be easier to show that it is a theorem.

Contrawise, there is no formal way of showing that a sentence is *not* a theorem. We would need to reason in English about all possible proofs. Yet there is a formal method for showing that a sentence is not a tautology. We need only construct a model in which the sentence is false. Given a choice between showing that a sentence is not a theorem and showing that it is not a tautology, it would be easier to show that it is not a tautology.

Fortunately, a sentence is a theorem if and only if it is a tautology. If we provide a proof of $\vdash \mathcal{A}$ and thus show that it is a theorem, it follows that $\mathcal{A}$ is a tautology; i.e., $\models \mathcal{A}$. Similarly, if we construct a model in which $\mathcal{A}$ is false and thus show that it is not a tautology, it follows that $\mathcal{A}$ is not a theorem.

In general, $\mathcal{A} \vdash \mathcal{B}$ if and only if $\mathcal{A} \models \mathcal{B}$. As such:

- ▷ An argument is *valid* if and only if *the conclusion is derivable from the premises.*
- ▷ Two sentences are *logically equivalent* if and only if they are *provably equivalent.*
- ▷ A set of sentences is *consistent* if and only if it is *not provably inconsistent.*

You can pick and choose when to think in terms of proofs and when to think in terms of models, doing whichever is easier for a given task. Table 7.1 summarizes when it is best to give proofs and when it is best to give models.

In this way, proofs and models give us a versatile toolkit for working with arguments. If we can translate an argument into QL, then we can measure its logical weight in a purely formal way. If it is deductively valid, we can give a formal proof; if it is invalid, we can provide a formal counterexample.

	YES	NO
Is $\mathcal{A}$ a tautology?	prove $\vdash \mathcal{A}$	give a model in which $\mathcal{A}$ is false
Is $\mathcal{A}$ a contradiction?	prove $\vdash \neg\mathcal{A}$	give a model in which $\mathcal{A}$ is true
Is $\mathcal{A}$ contingent?	give a model in which $\mathcal{A}$ is true and another in which $\mathcal{A}$ is false	prove $\vdash \mathcal{A}$ or $\vdash \neg\mathcal{A}$
Are $\mathcal{A}$ and $\mathcal{B}$ equivalent?	prove $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$	give a model in which $\mathcal{A}$ and $\mathcal{B}$ have different truth values
Is the set $\mathbb{A}$ consistent?	give a model in which all the sentences in $\mathbb{A}$ are true	taking the sentences in $\mathbb{A}$, prove $\mathcal{B}$ and $\neg\mathcal{B}$
Is the argument '$\mathcal{P}$, $\therefore$ $\mathcal{C}$' valid?	prove $\mathcal{P} \vdash \mathcal{C}$	give a model in which $\mathcal{P}$ is true and $\mathcal{C}$ is false

Table 7.1: Sometimes it is easier to show something by providing proofs than it is by providing models. Sometimes it is the other way round. It depends on what you are trying to show.

7.5 Soundness and completeness

This toolkit is incredibly convenient. It is also intuitive, because it seems natural that provability and semantic entailment should agree. Yet, do not be fooled by the similarity of the symbols '$\models$' and '$\vdash$.' The fact that these two are really interchangeable is not a simple thing to prove.

Why should we think that an argument that *can be proven* is necessarily a *valid* argument? That is, why think that $\mathcal{A} \vdash \mathcal{B}$ implies $\mathcal{A} \models \mathcal{B}$?

This is the problem of SOUNDNESS. A proof system is SOUND if there are no proofs of invalid arguments. Demonstrating that the proof system is sound would require showing that *any* possible proof is the proof of a valid argument. It would not be enough simply to succeed when trying to prove many valid arguments and to fail when trying to prove invalid ones.

Fortunately, there is a way of approaching this in a step-wise fashion. If using the $\wedge$E rule on the last line of a proof could never change a valid argument into an invalid one, then using the rule many times could not make an argument invalid. Similarly, if using the $\wedge$E and $\vee$E rules individually on the last line of a proof could never change a valid argument into an invalid one, then using them in combination could not either.

The strategy is to show for every rule of inference that it alone could not make a valid argument into an invalid one. It follows that the rules used in combination would not make a valid argument invalid. Since a proof is just a series of lines, each justified by a rule of inference, this would show that every provable argument is valid.

Consider, for example, the $\wedge$I rule. Suppose we use it to add $\mathcal{A}\wedge\mathcal{B}$ to a valid argument. In order for the rule to apply, $\mathcal{A}$ and $\mathcal{B}$ must already be available in the proof. Since the argument so far is valid, $\mathcal{A}$ and $\mathcal{B}$ are either premises of the argument or valid consequences of the premises. As such, any model in which the premises are true must be a model in which $\mathcal{A}$ and $\mathcal{B}$ are true. According to the definition of TRUTH IN QL, this means that $\mathcal{A}\wedge\mathcal{B}$ is also true in such a model. Therefore, $\mathcal{A}\wedge\mathcal{B}$ validly follows from the premises. This means that using the $\wedge$E rule to extend a valid proof produces another valid proof.

In order to show that the proof system is sound, we would need to show this for the other inference rules. Since the derived rules are consequences of the basic rules, it would suffice to provide similar arguments for the 16 other basic rules. This tedious exercise falls beyond the scope of this book.

Given a proof that the proof system is sound, it follows that every theorem is a tautology.

It is still possible to ask: Why think that *every* valid argument is an argument that can be proven? That is, why think that $\mathcal{A} \models \mathcal{B}$ implies $\mathcal{A} \vdash \mathcal{B}$?

This is the problem of COMPLETENESS. A proof system is COMPLETE if there is a proof of every valid argument. Completeness for a language like QL was first proven by Kurt Gödel in 1929. The proof is beyond the scope of this book.

The important point is that, happily, the proof system for QL is both sound and complete. This is not the case for all proof systems and all formal languages. Because it is true of QL, we can choose to give proofs or construct models—whichever is easier for the task at hand.

Summary of definitions

- ▷ A sentence $\mathcal{A}$ is a THEOREM if and only if $\vdash \mathcal{A}$.

- ▷ Two sentences $\mathcal{A}$ and $\mathcal{B}$ are PROVABLY EQUIVALENT if and only if $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$.

- ▷ $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\}$ is PROVABLY INCONSISTENT if and only if, for some sentence $\mathcal{B}$, $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\} \vdash (\mathcal{B} \wedge \neg\mathcal{B})$.

Practice Exercises

Part A

1. Without using the QN rule, prove $\neg\exists x\neg\mathcal{A} \vdash \forall x\mathcal{A}$

⋆ Part B

1. Identify which of the following are substitution instances of $\forall xRcx$: Rac, Rca, Raa, Rcb, Rbc, Rcc, Rcd, Rcx
2. Identify which of the following are substitution instances of $\exists x\forall yLxy$: $\forall yLby$, $\forall xLbx$, Lab, $\exists xLxa$

⋆ **Part C** Provide a justification (rule and line numbers) for each line of proof that requires one.

1	$\forall x \exists y(Rxy \lor Ryx)$
2	$\forall x \neg Rmx$
3	$\exists y(Rmy \lor Rym)$
4	$\quad\mid Rma \lor Ram$
5	$\quad\mid \neg Rma$
6	$\quad\mid Ram$
7	$\quad\mid \exists x Rxm$
8	$\exists x Rxm$

1	$\forall x(Jx \to Kx)$
2	$\exists x \forall y Lxy$
3	$\forall x Jx$
4	$\quad\mid \forall y Lay$
5	$\quad\mid Ja$
6	$\quad\mid Ja \to Ka$
7	$\quad\mid Ka$
8	$\quad\mid Laa$
9	$\quad\mid Ka \land Laa$
10	$\quad\mid \exists x(Kx \land Lxx)$
11	$\exists x(Kx \land Lxx)$

1	$\forall x(\exists y Lxy \to \forall z Lzx)$
2	Lab
3	$\exists y Lay \to \forall z Lza$
4	$\exists y Lay$
5	$\forall z Lza$
6	Lca
7	$\exists y Lcy \to \forall z Lzc$
8	$\exists y Lcy$
9	$\forall z Lzc$
10	Lcc
11	$\forall x Lxx$

1	$\quad\mid \neg(\exists x Mx \lor \forall x \neg Mx)$
2	$\quad\mid \neg\exists x Mx \land \neg\forall x \neg Mx$
3	$\quad\mid \neg\exists x Mx$
4	$\quad\mid \forall x \neg Mx$
5	$\quad\mid \neg\forall x \neg Mx$
6	$\exists x Mx \lor \forall x \neg Mx$

⋆ **Part D** Provide a proof of each claim.

1. $\vdash \forall x Fx \lor \neg\forall x Fx$
2. $\{\forall x(Mx \leftrightarrow Nx), Ma \land \exists x Rxa\} \vdash \exists x Nx$
3. $\{\forall x(\neg Mx \lor Ljx), \forall x(Bx \to Ljx), \forall x(Mx \lor Bx)\} \vdash \forall x Ljx$
4. $\forall x(Cx \land Dt) \vdash \forall x Cx \land Dt$
5. $\exists x(Cx \lor Dt) \vdash \exists x Cx \lor Dt$

Part E Provide a proof of the argument about Billy on p. 79.

Part F Look back at Part B on p. 90. Provide proofs to show that each of the argument forms is valid in QL.

Part G Aristotle and his successors identified other syllogistic forms. Symbolize each of the following argument forms in QL and add the additional assumptions 'There is an A' and 'There is a B.' Then prove that the supplemented arguments forms are valid in QL.

1. **Darapti:** All As are Bs. All As are Cs. $\therefore$ Some B is C.
2. **Felapton:** No Bs are Cs. All As are Bs. $\therefore$ Some A is not C.
3. **Barbari:** All Bs are Cs. All As are Bs. $\therefore$ Some A is C.
4. **Camestros:** All Cs are Bs. No As are Bs. $\therefore$ Some A is not C.
5. **Celaront:** No Bs are Cs. All As are Bs. $\therefore$ Some A is not C.
6. **Cesaro:** No Cs are Bs. All As are Bs. $\therefore$ Some A is not C.
7. **Fapesmo:** All Bs are Cs. No As are Bs. $\therefore$ Some C is not A.

Part H Provide a proof of each claim.

1. $\forall x \forall y Gxy \vdash \exists x Gxx$
2. $\forall x \forall y (Gxy \rightarrow Gyx) \vdash \forall x \forall y (Gxy \leftrightarrow Gyx)$
3. $\{\forall x(Ax \rightarrow Bx), \exists x Ax\} \vdash \exists x Bx$
4. $\{Na \rightarrow \forall x(Mx \leftrightarrow Ma), Ma, \neg Mb\} \vdash \neg Na$
5. $\vdash \forall z(Pz \vee \neg Pz)$
6. $\vdash \forall x Rxx \rightarrow \exists x \exists y Rxy$
7. $\vdash \forall y \exists x(Qy \rightarrow Qx)$

Part I Show that each pair of sentences is provably equivalent.

1. $\forall x(Ax \rightarrow \neg Bx)$, $\neg \exists x(Ax \wedge Bx)$
2. $\forall x(\neg Ax \rightarrow Bd)$, $\forall x Ax \vee Bd$
3. $\exists x Px \rightarrow Qc$, $\forall x(Px \rightarrow Qc)$

Part J Show that each of the following is provably inconsistent.

1. $\{Sa \rightarrow Tm, Tm \rightarrow Sa, Tm \wedge \neg Sa\}$
2. $\{\neg \exists x Rxa, \forall x \forall y Ryx\}$
3. $\{\neg \exists x \exists y Lxy, Laa\}$
4. $\{\forall x(Px \rightarrow Qx), \forall z(Pz \rightarrow Rz), \forall y Py, \neg Qa \wedge \neg Rb\}$

$\star$ **Part K** Write a symbolization key for the following argument, translate it, and prove it:

> There is someone who likes everyone who likes everyone that he likes. Therefore, there is someone who likes himself.

Part L Provide a proof of each claim.

1. $\{Pa \vee Qb, Qb \to b = c, \neg Pa\} \vdash Qc$
2. $\{m = n \vee n = o, An\} \vdash Am \vee Ao$
3. $\{\forall x x = m, Rma\} \vdash \exists x Rxx$
4. $\neg\exists x x \neq m \vdash \forall x \forall y(Px \to Py)$
5. $\forall x \forall y(Rxy \to x = y) \vdash Rab \to Rba$
6. $\{\exists x Jx, \exists x \neg Jx\} \vdash \exists x \exists y\ x \neq y$
7. $\{\forall x(x = n \leftrightarrow Mx), \forall x(Ox \vee \neg Mx)\} \vdash On$
8. $\{\exists x Dx, \forall x(x = p \leftrightarrow Dx)\} \vdash Dp$
9. $\{\exists x\big[Kx \wedge \forall y(Ky \to x = y) \wedge Bx\big], Kd\} \vdash Bd$
10. $\vdash Pa \to \forall x(Px \vee x \neq a)$

Part M Look back at Part D on p. 91. For each argument: If it is valid in QL, give a proof. If it is invalid, construct a model to show that it is invalid.

$\star$ **Part N** For each of the following pairs of sentences: If they are logically equivalent in QL, give proofs to show this. If they are not, construct a model to show this.

1. $\forall x Px \to Qc$, $\forall x(Px \to Qc)$
2. $\forall x Px \wedge Qc$, $\forall x(Px \wedge Qc)$
3. $Qc \vee \exists x Qx$, $\exists x(Qc \vee Qx)$
4. $\forall x \forall y \forall z Bxyz$, $\forall x Bxxx$
5. $\forall x \forall y Dxy$, $\forall y \forall x Dxy$
6. $\exists x \forall y Dxy$, $\forall y \exists x Dxy$

$\star$ **Part O** For each of the following arguments: If it is valid in QL, give a proof. If it is invalid, construct a model to show that it is invalid.

1. $\forall x \exists y Rxy$, $\therefore \exists y \forall x Rxy$
2. $\exists y \forall x Rxy$, $\therefore \forall x \exists y Rxy$
3. $\exists x(Px \wedge \neg Qx)$, $\therefore \forall x(Px \to \neg Qx)$
4. $\forall x(Sx \to Ta)$, Sd, $\therefore Ta$
5. $\forall x(Ax \to Bx)$, $\forall x(Bx \to Cx)$, $\therefore \forall x(Ax \to Cx)$
6. $\exists x(Dx \vee Ex)$, $\forall x(Dx \to Fx)$, $\therefore \exists x(Dx \wedge Fx)$
7. $\forall x \forall y(Rxy \vee Ryx)$, $\therefore Rjj$
8. $\exists x \exists y(Rxy \vee Ryx)$, $\therefore Rjj$
9. $\forall x Px \to \forall x Qx$, $\exists x \neg Px$, $\therefore \exists x \neg Qx$
10. $\exists x Mx \to \exists x Nx$, $\neg\exists x Nx$, $\therefore \forall x \neg Mx$

Part P

1. If you know that $\mathcal{A} \vdash \mathcal{B}$, what can you say about $(\mathcal{A} \wedge \mathcal{C}) \vdash \mathcal{B}$? Explain your answer.
2. If you know that $\mathcal{A} \vdash \mathcal{B}$, what can you say about $(\mathcal{A} \vee \mathcal{C}) \vdash \mathcal{B}$? Explain your answer.

Chapter A

Symbolic notation

In the history of formal logic, different symbols have been used at different times and by different authors. Often, authors were forced to use notation that their printers could typeset.

In one sense, the symbols used for various logical constants is arbitrary. There is nothing written in heaven that says that '$\neg$' must be the symbol for truth-functional negation. We might have specified a different symbol to play that part. Once we have given definitions for well-formed formulae (wff) and for truth in our logic languages, however, using '$\neg$' is no longer arbitrary. That is the symbol for negation in this textbook, and so it is the symbol for negation when writing sentences in our languages SL or QL.

This appendix presents some common symbols, so that you can recognize them if you encounter them in an article or in another book.

summary of symbols	
negation	$\neg$, $\sim$
conjunction	&, $\wedge$, $\bullet$
disjunction	$\vee$
conditional	$\rightarrow$, $\supset$
biconditional	$\leftrightarrow$, $\equiv$

Negation Two commonly used symbols are the *hoe*, '$\neg$', and the *swung dash*, '$\sim$.' In some more advanced formal systems it is necessary to distinguish between two kinds of negation; the distinction is sometimes represented by using both '$\neg$' and '$\sim$.'

Disjunction The symbol '$\vee$' is typically used to symbolize inclusive disjunction.

Conjunction Conjunction is often symbolized with the *ampersand*, '&.' The ampersand is actually a decorative form of the Latin word 'et' which means 'and'; it is commonly used in English writing. As a symbol in a formal system, the ampersand is not the word 'and'; its meaning is given by the formal semantics for the language. Perhaps to avoid this confusion, some systems use a different symbol for conjunction. For example, '$\wedge$' is a counterpart to the symbol used for disjunction. Sometimes a single dot, '$\bullet$', is used. In some older texts, there is no symbol for conjunction at all; 'A and B' is simply written 'AB.'

Material Conditional There are two common symbols for the material conditional: the *arrow*, '$\rightarrow$', and the *hook*, '$\supset$.'

Material Biconditional The *double-headed arrow*, '$\leftrightarrow$', is used in systems that use the arrow to represent the material conditional. Systems that use the hook for the conditional typically use the *triple bar*, '$\equiv$', for the biconditional.

Quantifiers The universal quantifier is typically symbolized as an upside-down A, '$\forall$', and the existential quantifier as a backwards E, '$\exists$.' In some texts, there is no separate symbol for the universal quantifier. Instead, the variable is just written in parentheses in front of the formula that it binds. For example, 'all x are P' is written $(x)Px$.

In some systems, the quantifiers are symbolized with larger versions of the symbols used for conjunction and disjunction. Although quantified expressions cannot be translated into expressions without quantifiers, there is a conceptual connection between the universal quantifier and conjunction and between the existential quantifier and disjunction. Consider the sentence $\exists x Px$, for example. It means that *either* the first member of the UD is a P, *or* the second one is, *or* the third one is, ... Such a system uses the symbol '$\bigvee$' instead of '$\exists$.'

Polish notation

This section briefly discusses sentential logic in Polish notation, a system of notation introduced in the late 1920s by the Polish logician Jan Łukasiewicz.

notation of SL	Polish notation
$\neg$	N
$\wedge$	K
$\vee$	A
$\rightarrow$	C
$\leftrightarrow$	E

Lower case letters are used as sentence letters. The capital letter N is used for negation. A is used for disjunction, K for conjunction, C for the conditional, E for the biconditional. ('A' is for alternation, another name for logical disjunction. 'E' is for equivalence.)

In Polish notation, a binary connective is written *before* the two sentences that it connects. For example, the sentence $A \wedge B$ of SL would be written Kab in Polish notation.

The sentences $\neg A \rightarrow B$ and $\neg(A \rightarrow B)$ are very different; the main logical operator of the first is the conditional, but the main connective of the second is negation. In SL, we show this by putting parentheses around the conditional in the second sentence. In Polish notation, parentheses are never required. The left-most connective is always the main connective. The first sentence would simply be written $CNab$ and the second $NCab$.

This feature of Polish notation means that it is possible to evaluate sentences simply by working through the symbols from right to left. If you were constructing a truth table for $NKab$, for example, you would first consider the truth-values assigned to b and a, then consider their conjunction, and then negate the result. The general rule for what to evaluate next in SL is not nearly so simple. In SL, the truth table for $\neg(A \wedge B)$ requires looking at A and B, then looking in the middle of the sentence at the conjunction, and then at the beginning of the sentence at the negation. Because the order of operations can be specified more mechanically in Polish notation, variants of Polish notation are used as the internal structure for many computer programming languages.

Chapter B

Solutions to selected exercises

Many of the exercises may be answered correctly in different ways. Where that is the case, the solution here represents one possible correct answer.

Chapter 1 Part C

1. consistent
2. inconsistent
3. consistent
4. consistent

Chapter 1 Part D 1, 2, 3, 6, 8, and 10 are possible.

Chapter 2 Part A

1. $\neg M$
2. $M \vee \neg M$
3. $G \vee C$
4. $\neg C \wedge \neg G$
5. $C \rightarrow (\neg G \wedge \neg M)$
6. $M \vee (C \vee G)$

Chapter 2 Part C

1. $E_1 \wedge E_2$
2. $F_1 \rightarrow S_1$
3. $F_1 \vee E_1$
4. $E_2 \wedge \neg S_2$
5. $\neg E_1 \wedge \neg E_2$
6. $E_1 \wedge E_2 \wedge \neg(S_1 \vee S_2)$

7. $S_2 \rightarrow F_2$
8. $(\neg E_1 \rightarrow \neg E_2) \wedge (E_1 \rightarrow E_2)$
9. $S_1 \leftrightarrow \neg S_2$
10. $(E_2 \wedge F_2) \rightarrow S_2$
11. $\neg(E_2 \wedge F_2)$
12. $(F_1 \wedge F_2) \leftrightarrow (\neg E_1 \wedge \neg E_2)$

Chapter 2 Part D

A: Alice is a spy.
B: Bob is a spy.
C: The code has been broken.
G: The German embassy will be in an uproar.

1. $A \wedge B$
2. $(A \vee B) \rightarrow C$
3. $\neg(A \vee B) \rightarrow \neg C$
4. $G \vee C$
5. $(C \vee \neg C) \wedge G$
6. $(A \vee B) \wedge \neg(A \wedge B)$

Chapter 2 Part G

1. (a) no (b) no
2. (a) no (b) yes
3. (a) yes (b) yes
4. (a) no (b) no
5. (a) yes (b) yes
6. (a) no (b) no
7. (a) no (b) yes
8. (a) no (b) yes
9. (a) no (b) no

Chapter 3 Part A

1. tautology
2. contradiction
3. contingent
4. tautology
5. tautology
6. contingent
7. tautology
8. contradiction
9. tautology
10. contradiction

11. tautology
12. contingent
13. contradiction
14. contingent
15. tautology
16. tautology
17. contingent
18. contingent

Chapter 3 Part B 2, 3, 5, 6, 8, and 9 are logically equivalent.

Chapter 3 Part C 1, 3, 6, 7, and 8 are consistent.

Chapter 3 Part D 3, 5, 8, and 10 are valid.

Chapter 3 Part E

1. $\mathcal{A}$ and $\mathcal{B}$ have the same truth value on every line of a complete truth table, so $\mathcal{A} \leftrightarrow \mathcal{B}$ is true on every line. It is a tautology.
2. The sentence is false on some line of a complete truth table. On that line, $\mathcal{A}$ and $\mathcal{B}$ are true and $\mathcal{C}$ is false. So the argument is invalid.
3. Since there is no line of a complete truth table on which all three sentences are true, the conjunction is false on every line. So it is a contradiction.
4. Since $\mathcal{A}$ is false on every line of a complete truth table, there is no line on which $\mathcal{A}$ and $\mathcal{B}$ are true and $\mathcal{C}$ is false. So the argument is valid.
5. Since $\mathcal{C}$ is true on every line of a complete truth table, there is no line on which $\mathcal{A}$ and $\mathcal{B}$ are true and $\mathcal{C}$ is false. So the argument is valid.
6. Not much. $(\mathcal{A} \vee \mathcal{B})$ is a tautology if $\mathcal{A}$ and $\mathcal{B}$ are tautologies; it is a contradiction if they are contradictions; it is contingent if they are contingent.
7. $\mathcal{A}$ and $\mathcal{B}$ have different truth values on at least one line of a complete truth table, and $(\mathcal{A} \vee \mathcal{B})$ will be true on that line. On other lines, it might be true or false. So $(\mathcal{A} \vee \mathcal{B})$ is either a tautology or it is contingent; it is *not* a contradiction.

Chapter 3 Part F

1. $\neg A \rightarrow B$
2. $\neg(A \rightarrow \neg B)$
3. $\neg[(A \rightarrow B) \rightarrow \neg(B \rightarrow A)]$

Chapter 4 Part A

1	$W \rightarrow \neg B$	
2	$A \wedge W$	
3	$B \vee (J \wedge K)$	
4	W	$\wedge$E 2
5	$\neg B$	$\rightarrow$E 1, 4
6	$J \wedge K$	$\vee$E 3, 5
7	K	$\wedge$E 6

1	$L \leftrightarrow \neg O$	
2	$L \vee \neg O$	
3	$\quad \neg L$	
4	$\quad \neg O$	$\vee$E 2, 3
5	$\quad L$	$\leftrightarrow$E 1, 4
6	$\quad \neg L$	R 3
7	L	$\neg$E 3–6

1	$Z \to (C \wedge \neg N)$	
2	$\neg Z \to (N \wedge \neg C)$	
3	$\quad \neg(N \vee C)$	
4	$\quad \neg N \wedge \neg C$	DeM 3
5	$\quad\quad Z$	
6	$\quad\quad C \wedge \neg N$	$\to$E 1, 5
7	$\quad\quad C$	$\wedge$E 6
8	$\quad\quad \neg C$	$\wedge$E 4
9	$\quad \neg Z$	$\neg$I 5–8
10	$\quad N \wedge \neg C$	$\to$E 2, 9
11	$\quad N$	$\wedge$E 10
12	$\quad \neg N$	$\wedge$E 4
13	$N \vee C$	$\neg$E 3–12

Chapter 4 Part B

1.

1	$K \wedge L$	want $K \leftrightarrow L$
2	$\quad K$	want L
3	$\quad L$	$\wedge$E 1
4	$\quad L$	want K
5	$\quad K$	$\wedge$E 1
6	$K \leftrightarrow L$	$\leftrightarrow$I 2–3, 4–5

2.

1	$A \to (B \to C)$	want $(A \wedge B) \to C$
2	$\quad A \wedge B$	want C
3	$\quad A$	$\wedge$E 2
4	$\quad B \to C$	$\to$E 1, 3
5	$\quad B$	$\wedge$E 2
6	$\quad C$	$\to$E 4, 5
7	$(A \wedge B) \to C$	$\to$I 2–6

3.

1	$P \wedge (Q \vee R)$	
2	$P \to \neg R$	want $Q \vee E$
3	P	$\wedge$E 1
4	$\neg R$	$\to$E 2, 3
5	$Q \vee R$	$\wedge$E 1
6	Q	$\vee$E 5, 4
7	$Q \vee E$	$\vee$I 6

4.

1	$(C \wedge D) \vee E$	want $E \vee D$
2	$\quad \neg E$	want D
3	$\quad C \wedge D$	$\vee$E 1, 2
4	$\quad D$	$\wedge$E 3
5	$\neg E \to D$	$\to$I 2–4
6	$E \vee D$	MC 5

5.

1	$\neg F \to G$	
2	$F \to H$	want $G \vee H$
3	$\quad \neg G$	want H
4	$\quad \neg\neg F$	MT 1, 3
5	$\quad F$	DN 4
6	$\quad H$	$\to$E 2, 5
7	$\neg G \to H$	$\to$I 3–6
8	$G \vee H$	MC 7

6.

1	$(X \wedge Y) \vee (X \wedge Z)$	
2	$\neg(X \wedge D)$	
3	$D \vee M$	want M
4	$\neg X$	for reductio
5	$\neg X \vee \neg Y$	$\vee$I 4
6	$\neg(X \wedge Y)$	DeM 5
7	$X \wedge Z$	$\vee$E 1, 6
8	X	$\wedge$E 7
9	$\neg X$	R 4
10	X	$\neg$E 4–9
11	$\neg M$	for reductio
12	D	$\vee$E 3, 11
13	$X \wedge D$	$\wedge$I 10, 12
14	$\neg(X \wedge D)$	R 2
15	M	$\neg$E 11–14

Chapter 5 Part A

1. $Za \wedge Zb \wedge Zc$
2. $Rb \wedge \neg Ab$
3. $Lcb \to Mb$
4. $(Ab \wedge Ac) \to (Lab \wedge Lac)$
5. $\exists x(Rx \wedge Zx)$
6. $\forall x(Ax \to Rx)$
7. $\forall x\big[Zx \to (Mx \vee Ax)\big]$
8. $\exists x(Rx \wedge \neg Ax)$
9. $\exists x(Rx \wedge Lcx)$
10. $\forall x\big[(Mx \wedge Zx) \to Lbx\big]$
11. $\forall x\big[(Mx \wedge Lax) \to Lxa\big]$
12. $\exists x Rx \to Ra$
13. $\forall x(Ax \to Rx)$
14. $\forall x\big[(Mx \wedge Lcx) \to Lax\big]$
15. $\exists x(Mx \wedge Lxb \wedge \neg Lbx)$

Chapter 5 Part E

1. $\neg\exists x Tx$

2. $\forall x(Mx \to Sx)$
3. $\exists x \neg Sx$
4. $\exists x[Cx \wedge \neg \exists y Byx]$
5. $\neg \exists x Bxx$
6. $\neg \exists x(Cx \wedge \neg Sx \wedge Tx)$
7. $\exists x(Cx \wedge Tx) \wedge \exists x(Mx \wedge Tx) \wedge \neg \exists x(Cx \wedge Mx \wedge Tx)$
8. $\forall x[Cx \to \forall y(\neg Cy \to Bxy)]$
9. $\forall x\big((Cx \wedge Mx) \to \forall y[(\neg Cy \wedge \neg My) \to Bxy]\big)$

Chapter 5 Part G

1. $\forall x(Cxp \to Dx)$
2. $Cjp \wedge Fj$
3. $\exists x(Cxp \wedge Fx)$
4. $\neg \exists x Sxj$
5. $\forall x\big[(Cxp \wedge Fx) \to Dx\big]$
6. $\neg \exists x(Cxp \wedge Mx)$
7. $\exists x(Cjx \wedge Sxe \wedge Fj)$
8. $Spe \wedge Mp$
9. $\forall x\big[(Sxp \wedge Mx) \to \neg \exists y Cyx\big]$
10. $\exists x(Sxj \wedge \exists y Cyx \wedge Fj)$
11. $\forall x\big[Dx \to \exists y(Sxy \wedge Fy \wedge Dy)\big]$
12. $\forall x\big[(Mx \wedge Dx) \to \exists y(Cxy \wedge Dy)\big]$

Chapter 5 Part J

1. $\forall x(Cx \to Bx)$
2. $\neg \exists x Wx$
3. $\exists x \exists y(Cx \wedge Cy \wedge x \neq y)$
4. $\exists x \exists y(Jx \wedge Ox \wedge Jy \wedge Oy \wedge x \neq y)$
5. $\forall x \forall y \forall z\big[(Jx \wedge Ox \wedge Jy \wedge Oy \wedge Jz \wedge Oz) \to (x = y \vee x = z \vee y = z)\big]$
6. $\exists x \exists y\big(Jx \wedge Bx \wedge Jy \wedge By \wedge x \neq y \wedge \forall z[(Jz \wedge Bz) \to (x = z \vee y = z)]\big)$
7. $\exists x_1 \exists x_2 \exists x_3 \exists x_4\big[Dx_1 \wedge Dx_2 \wedge Dx_3 \wedge Dx_4 \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4 \wedge \neg \exists y(Dy \wedge y \neq x_1 \wedge y \neq x_2 \wedge y \neq x_3 \wedge y \neq x_4)\big]$
8. $\exists x\big(Dx \wedge Cx \wedge \forall y[(Dy \wedge Cy) \to x = y] \wedge Bx\big)$
9. $\forall x\big[(Ox \wedge Jx) \to Wx\big] \wedge \exists x\big[Mx \wedge \forall y(My \to x = y) \wedge Wx\big]$
10. $\exists x\big(Dx \wedge Cx \wedge \forall y[(Dy \wedge Cy) \to x = y] \wedge Wx\big) \to \exists x \forall y(Wx \leftrightarrow x = y)$
11. wide scope: $\neg \exists x\big[Mx \wedge \forall y(My \to x = y) \wedge Jx\big]$
    narrow scope: $\exists x\big[Mx \wedge \forall y(My \to x = y) \wedge \neg Jx\big]$
12. wide scope: $\neg \exists x \exists z\big(Dx \wedge Cx \wedge Mz \wedge \forall y[(Dy \wedge Cy) \to x = y] \wedge \forall y[(My \to z = y) \wedge x = z]\big)$
    narrow scope: $\exists x \exists z\big(Dx \wedge Cx \wedge Mz \wedge \forall y[(Dy \wedge Cy) \to x = y] \wedge \forall y[(My \to z = y) \wedge x \neq z]\big)$

Chapter 6 Part A 2, 3, 4, 6, 8, and 9 are true in the model.

Chapter 6 Part B 4, 5, and 7 are true in the model.

Chapter 6 Part D

$$\begin{aligned} \text{UD} &= \{10,11,12,13\} \\ \text{extension}(O) &= \{11,13\} \\ \text{extension}(S) &= \emptyset \\ \text{extension}(T) &= \{10,11,12,13\} \\ \text{extension}(U) &= \{13\} \\ \text{extension}(N) &= \{<11,10>,<12,11>,<13,12>\} \end{aligned}$$

Chapter 6 Part E

1. The sentence is true in this model:
$$\begin{aligned} \text{UD} &= \{\text{Stan}\} \\ \text{extension}(D) &= \{\text{Stan}\} \\ \text{referent}(a) &= \text{Stan} \\ \text{referent}(b) &= \text{Stan} \end{aligned}$$
And it is false in this model:
$$\begin{aligned} \text{UD} &= \{\text{Stan}\} \\ \text{extension}(D) &= \emptyset \\ \text{referent}(a) &= \text{Stan} \\ \text{referent}(b) &= \text{Stan} \end{aligned}$$
2. The sentence is true in this model:
$$\begin{aligned} \text{UD} &= \{\text{Stan}\} \\ \text{extension}(T) &= \{<\text{Stan, Stan}>\} \\ \text{referent}(h) &= \text{Stan} \end{aligned}$$
And it is false in this model:
$$\begin{aligned} \text{UD} &= \{\text{Stan}\} \\ \text{extension}(T) &= \emptyset \\ \text{referent}(h) &= \text{Stan} \end{aligned}$$
3. The sentence is true in this model:
$$\begin{aligned} \text{UD} &= \{\text{Stan, Ollie}\} \\ \text{extension}(P) &= \{\text{Stan}\} \\ \text{referent}(m) &= \text{Stan} \end{aligned}$$
And it is false in this model:
$$\begin{aligned} \text{UD} &= \{\text{Stan}\} \\ \text{extension}(P) &= \emptyset \\ \text{referent}(m) &= \text{Stan} \end{aligned}$$

Chapter 6 Part F There are many possible correct answers. Here are some:

1. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{alpha}\}\\ \text{extension}(J) &= \{\text{alpha}\}\\ \text{extension}(K) &= \emptyset\\ \text{referent}(a) &= \text{alpha}\end{aligned}$$

2. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(J) &= \{\text{alpha}\}\\ \text{referent}(m) &= \text{omega}\end{aligned}$$

3. Making the first sentence false and the second true:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(R) &= \{<\text{alpha,alpha}>\}\end{aligned}$$

4. Making the first sentence false and the second true:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(P) &= \{\text{alpha}\}\\ \text{extension}(Q) &= \emptyset\\ \text{referent}(c) &= \text{alpha}\end{aligned}$$

5. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{iota}\}\\ \text{extension}(P) &= \emptyset\\ \text{extension}(Q) &= \emptyset\end{aligned}$$

6. Making the first sentence false and the second true:

$$\begin{aligned}\text{UD} &= \{\text{iota}\}\\ \text{extension}(P) &= \emptyset\\ \text{extension}(Q) &= \{\text{iota}\}\end{aligned}$$

7. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{iota}\}\\ \text{extension}(P) &= \emptyset\\ \text{extension}(Q) &= \{\text{iota}\}\end{aligned}$$

8. Making the first sentence true and the second false:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(R) &= \{<\text{alpha, omega}>, <\text{omega, alpha}>\}\end{aligned}$$

9. Making the first sentence false and the second true:

$$\begin{aligned}\text{UD} &= \{\text{alpha, omega}\}\\ \text{extension}(R) &= \{<\text{alpha, alpha}>, <\text{alpha, omega}>\}\end{aligned}$$

Chapter 6 Part I

1. There are many possible answers. Here is one:

$$\begin{aligned}\text{UD} &= \{\text{Harry, Sally}\}\\ \text{extension}(R) &= \{<\text{Sally, Harry}>\}\\ \text{referent}(a) &= \text{Harry}\end{aligned}$$

2. There are no predicates or constants, so we only need to give a UD. Any UD with 2 members will do.

3. We need to show that it is impossible to construct a model in which these are both true. Suppose $\exists x\, x \neq a$ is true in a model. There is something in the universe of discourse that is *not* the referent of a. So there are at least two things in the universe of discourse: referent(a) and this other thing. Call this other thing β— we know $a \neq \beta$. But if $a \neq \beta$, then $\forall x \forall y\, x = y$ is false. So the first sentence must be false if the second sentence is true. As such, there is no model in which they are both true. Therefore, they are inconsistent.

Chapter 7 Part B

1. Rca, Rcb, Rcc, and Rcd are substitution instances of $\forall x Rcx$.
2. Of the expressions listed, only $\forall y Lby$ is a substitution instance of $\exists x \forall y Lxy$.

Chapter 7 Part C

1	$\forall x \exists y (Rxy \lor Ryx)$	
2	$\forall x \neg Rmx$	
3	$\exists y (Rmy \lor Rym)$	∀E 1
4	$Rma \lor Ram$	
5	$\neg Rma$	∀E 2
6	Ram	∨E 4, 5
7	$\exists x Rxm$	∃I 6
8	$\exists x Rxm$	∃E 3, 4–7

1	$\forall x (Jx \to Kx)$	
2	$\exists x \forall y Lxy$	
3	$\forall x Jx$	
4	$\forall y Lay$	
5	Ja	∀E 3
6	$Ja \to Ka$	∀E 1
7	Ka	→E 6, 5
8	Laa	∀E 4
9	$Ka \land Laa$	∧I 7, 8
10	$\exists x (Kx \land Lxx)$	∃I 9
11	$\exists x (Kx \land Lxx)$	∃E 2, 4–10

1	$\forall x (\exists y Lxy \to \forall z Lzx)$	
2	Lab	
3	$\exists y Lay \to \forall z Lza$	∀E 1
4	$\exists y Lay$	∃I 2
5	$\forall z Lza$	→E 3, 4
6	Lca	∀E 5
7	$\exists y Lcy \to \forall z Lzc$	∀E 1
8	$\exists y Lcy$	∃I 6
9	$\forall z Lzc$	→E 7, 8
10	Lcc	∀E 9
11	$\forall x Lxx$	∀I 10

1	$\neg(\exists x Mx \lor \forall x \neg Mx)$	
2	$\neg \exists x Mx \land \neg \forall x \neg Mx$	DeM 1
3	$\neg \exists x Mx$	∧E 2
4	$\forall x \neg Mx$	QN 3
5	$\neg \forall x \neg Mx$	∧E 2
6	$\exists x Mx \lor \forall x \neg Mx$	¬E 1–5

Chapter 7 Part D

1.

1	$\neg(\forall x Fx \vee \neg\forall x Fx)$	for reductio
2	$\neg\forall x Fx \wedge \neg\neg\forall x Fx$	DeM 1
3	$\neg\forall x Fx$	$\wedge$E 2
4	$\neg\neg\forall x Fx$	$\wedge$E 2
5	$\forall x Fx \vee \neg\forall x Fx$	$\neg$E 1–4

2.

1	$\forall x(Mx \leftrightarrow Nx)$	
2	$Ma \wedge \exists x Rxa$	want $\exists x Nx$
3	$Ma \leftrightarrow Na$	$\forall$E 1
4	Ma	$\wedge$E 2
5	Na	$\leftrightarrow$E 3, 4
6	$\exists x Nx$	$\exists$I 5

3.

1	$\forall x(\neg Mx \vee Ljx)$	
2	$\forall x(Bx \to Ljx)$	
3	$\forall x(Mx \vee Bx)$	want $\forall x Ljx$
4	$\neg Ma \vee Lja$	$\forall$E 1
5	$Ma \to Lja$	MC 4
6	$Ba \to Lja$	$\forall$E 2
7	$Ma \vee Ba$	$\forall$E 3
8	Lja	DIL 7, 5, 6
9	$\forall x Ljx$	$\forall$I 8

4.

1	$\forall x(Cx \wedge Dt)$	want $\forall x Cx \wedge Dt$
2	$Ca \wedge Dt$	$\forall$E 1
3	Ca	$\wedge$E 2
4	$\forall x Cx$	$\forall$I 3
5	Dt	$\wedge$E 2
6	$\forall x Cx \wedge Dt$	$\wedge$I 4, 5

5.

1	$\exists x(Cx \vee Dt)$	want $\exists xCx \vee Dt$
2	$\quad Ca \vee Dt$	for $\exists$E
3	$\quad\quad \neg(\exists xCx \vee Dt)$	for reductio
4	$\quad\quad \neg\exists xCx \wedge \neg Dt$	DeM 3
5	$\quad\quad \neg Dt$	$\wedge$E 4
6	$\quad\quad Ca$	$\vee$E 2, 5
7	$\quad\quad \exists xCx$	$\exists$I 6
8	$\quad\quad \neg\exists xCx$	$\wedge$E 4
9	$\quad \exists xCx \vee Dt$	$\neg$E 3–8
10	$\exists xCx \vee Dt$	$\exists$E 1, 2–9

Chapter 7 Part K Regarding the translation of this argument, see p. 81.

1	$\exists x\forall y[\forall z(Lxz \to Lyz) \to Lxy]$	
2	$\quad \forall y[\forall z(Laz \to Lyz) \to Lay]$	
3	$\quad \forall z(Laz \to Laz) \to Laa$	$\forall$E 2
4	$\quad\quad \neg\exists xLxx$	for reductio
5	$\quad\quad \forall x\neg Lxx$	QN 4
6	$\quad\quad \neg Laa$	$\forall$E 5
7	$\quad\quad \neg\forall z(Laz \to Laz)$	MT 5, 6
8	$\quad\quad\quad Lab$	
9	$\quad\quad\quad Lab$	R 8
10	$\quad\quad Lab \to Lab$	$\to$I 8—9
11	$\quad\quad \forall z(Laz \to Laz)$	$\forall$I 10
12	$\quad\quad \neg\forall z(Laz \to Laz)$	R 7
13	$\quad \exists xLxx$	$\neg$E 4—12
14	$\exists xLxx$	$\exists$E 1, 2—13

Chapter 7 Part N 2, 3, and 5 are logically equivalent.

Chapter 7 Part O 2, 4, 5, 7, and 10 are valid. Here are complete answers for some of them:

1. UD = {mocha, freddo}
   extension(R) = {<mocha, freddo>, <freddo, mocha>}

2.

1	$\exists y\forall xRxy$	want $\forall x\exists yRxy$
2	$\quad \forall xRxa$	
3	$\quad Rba$	$\forall$E 2
4	$\quad \exists yRby$	$\exists$I 3
5	$\quad \forall x\exists yRxy$	$\forall$I 4
6	$\forall x\exists yRxy$	$\exists$E 1, 2–5

Quick Reference

$\mathcal{A}$	$\neg\mathcal{A}$
T	F
F	T

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\wedge\mathcal{B}$	$\mathcal{A}\vee\mathcal{B}$	$\mathcal{A}\rightarrow\mathcal{B}$	$\mathcal{A}\leftrightarrow\mathcal{B}$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

$\mathcal{A}$	$\neg\mathcal{A}$
1	0
0	1

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}\wedge\mathcal{B}$	$\mathcal{A}\vee\mathcal{B}$	$\mathcal{A}\rightarrow\mathcal{B}$	$\mathcal{A}\leftrightarrow\mathcal{B}$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	0	1	1	0
0	0	0	0	1	1

Symbolization

Sentential Connectives (chapter 2)

It is not the case that P.	$\neg P$
Either P, or Q.	$(P \vee Q)$
Neither P, nor Q.	$\neg(P \vee Q)$ or $(\neg P \wedge \neg Q)$
Both P, and Q.	$(P \wedge Q)$
If P, then Q.	$(P \rightarrow Q)$
P only if Q.	$(P \rightarrow Q)$
P if and only if Q.	$(P \leftrightarrow Q)$
Unless P, Q. P unless Q.	$(P \vee Q)$

Predicates (chapter 5)

All Fs are Gs.	$\forall x(Fx \rightarrow Gx)$
Some Fs are Gs.	$\exists x(Fx \wedge Gx)$
Not all Fs are Gs.	$\neg\forall x(Fx \rightarrow Gx)$ or $\exists x(Fx \wedge \neg Gx)$
No Fs are Gs.	$\forall x(Fx \rightarrow \neg Gx)$ or $\neg\exists x(Fx \wedge Gx)$

Identity (section 5.6)

Only j is G.	$\forall x(Gx \leftrightarrow x = j)$
Everything besides j is G.	$\forall x(x \neq j \rightarrow Gx)$
The F is G.	$\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge Gx)$

'The F is not G' can be translated two ways:

It is not the case that the F is G. (wide)	$\neg\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge Gx)$
The F is non-G. (narrow)	$\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge \neg Gx)$

Using identity to symbolize quantities

There are at least ______ Fs.

one $\exists x Fx$
two $\exists x_1 \exists x_2 (Fx_1 \wedge Fx_2 \wedge x_1 \neq x_2)$
three $\exists x_1 \exists x_2 \exists x_3 (Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3)$
four $\exists x_1 \exists x_2 \exists x_3 \exists x_4 (Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge Fx_4 \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge x_3 \neq x_4)$
n $\exists x_1 \cdots \exists x_n (Fx_1 \wedge \cdots \wedge Fx_n \wedge x_1 \neq x_2 \wedge \cdots \wedge x_{n-1} \neq x_n)$

There are at most ______ Fs.

One way to say 'at most n things are F' is to put a negation sign in front of one of the symbolizations above and say $\neg$'at least $n+1$ things are F.' Equivalently:

one $\forall x_1 \forall x_2 \big[(Fx_1 \wedge Fx_2) \rightarrow x_1 = x_2\big]$
two $\forall x_1 \forall x_2 \forall x_3 \big[(Fx_1 \wedge Fx_2 \wedge Fx_3) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3)\big]$
three $\forall x_1 \forall x_2 \forall x_3 \forall x_4 \big[(Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge Fx_4) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_1 = x_4 \vee x_2 = x_3 \vee x_2 = x_4 \vee x_3 = x_4)\big]$
n $\forall x_1 \cdots \forall x_{n+1} \big[(Fx_1 \wedge \cdots \wedge Fx_{n+1}) \rightarrow (x_1 = x_2 \vee \cdots \vee x_n = x_{n+1})\big]$

There are exactly ______ Fs.

One way to say 'exactly n things are F' is to conjoin two of the symbolizations above and say 'at least n things are F' $\wedge$ 'at most n things are F.' The following equivalent formulae are shorter:

zero $\forall x \neg Fx$
one $\exists x \big[Fx \wedge \neg \exists y (Fy \wedge x \neq y)\big]$
two $\exists x_1 \exists x_2 \big[Fx_1 \wedge Fx_2 \wedge x_1 \neq x_2 \wedge \neg \exists y \big(Fy \wedge y \neq x_1 \wedge y \neq x_2\big)\big]$
three $\exists x_1 \exists x_2 \exists x_3 \big[Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \wedge \neg \exists y (Fy \wedge y \neq x_1 \wedge y \neq x_2 \wedge y \neq x_3)\big]$
n $\exists x_1 \cdots \exists x_n \big[Fx_1 \wedge \cdots \wedge Fx_n \wedge x_1 \neq x_2 \wedge \cdots \wedge x_{n-1} \neq x_n \wedge \neg \exists y (Fy \wedge y \neq x_1 \wedge \cdots \wedge y \neq x_n)\big]$

Specifying the size of the UD

Removing F from the symbolizations above produces sentences that talk about the size of the UD. For instance, 'there are at least 2 things (in the UD)' may be symbolized as $\exists x \exists y (x \neq y)$.

Sometimes it is easier to show something by providing proofs than it is by providing models. Sometimes it is the other way round.

	YES	NO
Is $\mathcal{A}$ a tautology?	prove $\vdash \mathcal{A}$	give a model in which $\mathcal{A}$ is false
Is $\mathcal{A}$ a contradiction?	prove $\vdash \neg\mathcal{A}$	give a model in which $\mathcal{A}$ is true
Is $\mathcal{A}$ contingent?	give a model in which $\mathcal{A}$ is true and another in which $\mathcal{A}$ is false	prove $\vdash \mathcal{A}$ or $\vdash \neg\mathcal{A}$
Are $\mathcal{A}$ and $\mathcal{B}$ equivalent?	prove $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$	give a model in which $\mathcal{A}$ and $\mathcal{B}$ have different truth values
Is the set $\mathbb{A}$ consistent?	give a model in which all the sentences in $\mathbb{A}$ are true	taking the sentences in $\mathbb{A}$, prove $\mathcal{B}$ and $\neg\mathcal{B}$
Is the argument '$\mathcal{P}$, $\therefore$ $\mathcal{C}$' valid?	prove $\mathcal{P} \vdash \mathcal{C}$	give a model in which $\mathcal{P}$ is true and $\mathcal{C}$ is false

Basic Rules of Proof

Reiteration

m	$\mathcal{A}$	
	$\mathcal{A}$	R m

Conjunction Introduction

m	$\mathcal{A}$	
n	$\mathcal{B}$	
	$\mathcal{A} \wedge \mathcal{B}$	$\wedge$I m, n

Conjunction Elimination

m	$\mathcal{A} \wedge \mathcal{B}$	
	$\mathcal{A}$	$\wedge$E m

m	$\mathcal{A} \wedge \mathcal{B}$	
	$\mathcal{B}$	$\wedge$E m

Disjunction Introduction

m	$\mathcal{A}$	
	$\mathcal{A} \vee \mathcal{B}$	$\vee$I m

m	$\mathcal{A}$	
	$\mathcal{B} \vee \mathcal{A}$	$\vee$I m

Disjunction Elimination

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{B}$	
	$\mathcal{A}$	$\vee$E m, n

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{A}$	
	$\mathcal{B}$	$\vee$E m, n

Conditional Introduction

m	$\quad\mathcal{A}$	want $\mathcal{B}$
n	$\quad\mathcal{B}$	
	$\mathcal{A} \rightarrow \mathcal{B}$	$\rightarrow$I m–n

Conditional Elimination

m	$\mathcal{A} \rightarrow \mathcal{B}$	
n	$\mathcal{A}$	
	$\mathcal{B}$	$\rightarrow$E m, n

Biconditional Introduction

m	$\quad\mathcal{A}$	want $\mathcal{B}$
n	$\quad\mathcal{B}$	
p	$\quad\mathcal{B}$	want $\mathcal{A}$
q	$\quad\mathcal{A}$	
	$\mathcal{A} \leftrightarrow \mathcal{B}$	$\leftrightarrow$I m–n, p–q

Biconditional Elimination

m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{B}$	
	$\mathcal{A}$	$\leftrightarrow$E m, n

m	$\mathcal{A} \leftrightarrow \mathcal{B}$	
n	$\mathcal{A}$	
	$\mathcal{B}$	$\leftrightarrow$E m, n

Negation Introduction

m	$\quad\mathcal{A}$	for reductio
$n-1$	$\quad\mathcal{B}$	
n	$\quad\neg\mathcal{B}$	
	$\neg\mathcal{A}$	$\neg$I m–n

Negation Elimination

m	$\quad\neg\mathcal{A}$	for reductio
$n-1$	$\quad\mathcal{B}$	
n	$\quad\neg\mathcal{B}$	
	$\mathcal{A}$	$\neg$E m–n

Quantifier Rules

Existential Introduction

$$\begin{array}{l|ll} m & \mathcal{A}\mathcal{c} & \\ & \exists \mathcal{x}\mathcal{A}\mathcal{x} & \exists\text{I } m \end{array}$$

Note that $\mathcal{x}$ may replace some or all occurrences of $\mathcal{c}$ in $\mathcal{A}\mathcal{c}$.

Existential Elimination

$$\begin{array}{l|ll} m & \exists \mathcal{x}\mathcal{A}\mathcal{x} & \\ n & \quad \mathcal{A}\mathcal{c}^* & \\ p & \quad \mathcal{B} & \\ & \mathcal{B} & \exists\text{E } m,\ n\text{–}p \end{array}$$

* $\mathcal{c}$ must not appear in $\exists \mathcal{x}\mathcal{A}\mathcal{x}$, in $\mathcal{B}$, or in any undischarged assumption.

Universal Introduction

$$\begin{array}{l|ll} m & \mathcal{A}\mathcal{c}^* & \\ & \forall \mathcal{x}\mathcal{A}\mathcal{x} & \forall\text{I } m \end{array}$$

* $\mathcal{c}$ must not occur in any undischarged assumptions.

Universal Elimination

$$\begin{array}{l|ll} m & \forall \mathcal{x}\mathcal{A}\mathcal{x} & \\ & \mathcal{A}\mathcal{c} & \forall\text{E } m \end{array}$$

Identity Rules

$$\begin{array}{l|ll} & \mathcal{c} = \mathcal{c} & =\text{I} \end{array}$$

$$\begin{array}{l|ll} m & \mathcal{c} = \mathcal{d} & \\ n & \mathcal{A} & \\ & \mathcal{A}\mathcal{c} \circlearrowleft d & =\text{E } m,\ n \end{array}$$

One constant may replace some or all occurrences of the other.

Derived Rules

Dilemma

$$\begin{array}{l|ll} m & \mathcal{A} \vee \mathcal{B} & \\ n & \mathcal{A} \to \mathcal{C} & \\ p & \mathcal{B} \to \mathcal{C} & \\ & \mathcal{C} & \text{DIL } m,\ n,\ p \end{array}$$

Constructive Dilemma

$$\begin{array}{l|ll} m & \mathcal{A} \to \mathcal{B} & \\ n & \mathcal{C} \to \mathcal{D} & \\ p & \mathcal{A} \vee \mathcal{C} & \\ & \mathcal{B} \vee \mathcal{D} & \text{CD } m,\ n,\ p \end{array}$$

Modus Tollens

$$\begin{array}{l|ll} m & \mathcal{A} \to \mathcal{B} & \\ n & \neg\mathcal{B} & \\ & \neg\mathcal{A} & \text{MT } m,\ n \end{array}$$

Hypothetical Syllogism

$$\begin{array}{l|ll} m & \mathcal{A} \to \mathcal{B} & \\ n & \mathcal{B} \to \mathcal{C} & \\ & \mathcal{A} \to \mathcal{C} & \text{HS } m,\ n \end{array}$$

Replacement Rules

Commutivity (Comm)

$$(\mathcal{A} \wedge \mathcal{B}) \Longleftrightarrow (\mathcal{B} \wedge \mathcal{A})$$
$$(\mathcal{A} \vee \mathcal{B}) \Longleftrightarrow (\mathcal{B} \vee \mathcal{A})$$
$$(\mathcal{A} \leftrightarrow \mathcal{B}) \Longleftrightarrow (\mathcal{B} \leftrightarrow \mathcal{A})$$

DeMorgan (DeM)

$$\neg(\mathcal{A} \vee \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \wedge \neg\mathcal{B})$$
$$\neg(\mathcal{A} \wedge \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \vee \neg\mathcal{B})$$

Double Negation (DN)

$$\neg\neg\mathcal{A} \Longleftrightarrow \mathcal{A}$$

Material Conditional (MC)

$$(\mathcal{A} \to \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \vee \mathcal{B})$$
$$(\mathcal{A} \vee \mathcal{B}) \Longleftrightarrow (\neg\mathcal{A} \to \mathcal{B})$$

Biconditional Exchange ($\leftrightarrow$ex)

$$[(\mathcal{A} \to \mathcal{B}) \wedge (\mathcal{B} \to \mathcal{A})] \Longleftrightarrow (\mathcal{A} \leftrightarrow \mathcal{B})$$

Replacement Rules (contd.)

Quantifier Negation (QN)

$$\neg\forall\mathcal{X}\mathcal{A} \iff \exists\mathcal{X}\neg\mathcal{A}$$
$$\neg\exists\mathcal{X}\mathcal{A} \iff \forall\mathcal{X}\neg\mathcal{A}$$

Transposition (Transp)

$$(\mathcal{A} \to \mathcal{B}) \iff (\neg\mathcal{B} \to \neg\mathcal{A})$$

Associativity (Assc)

$$(\mathcal{A} \lor (\mathcal{B} \lor \mathcal{C})) \iff ((\mathcal{A} \lor \mathcal{B}) \lor \mathcal{C})$$
$$(\mathcal{A} \land (\mathcal{B} \land \mathcal{C})) \iff ((\mathcal{A} \land \mathcal{B}) \land \mathcal{C})$$

Distribution (Dist)

$$(\mathcal{A} \land (\mathcal{B} \lor \mathcal{C})) \iff ((\mathcal{A} \land \mathcal{B}) \lor (\mathcal{A} \land \mathcal{C}))$$
$$(\mathcal{A} \lor (\mathcal{B} \land \mathcal{C})) \iff ((\mathcal{A} \lor \mathcal{B}) \land (\mathcal{A} \lor \mathcal{C}))$$

Tautology (Taut)

$$(\mathcal{A} \lor \mathcal{A}) \iff \mathcal{A}$$
$$(\mathcal{A} \land \mathcal{A}) \iff \mathcal{A}$$

Exportation (Exp)

$$((\mathcal{A} \land \mathcal{B}) \to \mathcal{C}) \iff (\mathcal{A} \to (\mathcal{B} \to \mathcal{C}))$$

In the Introduction to his volume *Symbolic Logic*, Charles Lutwidge Dodson advised: “When you come to any passage you don’t understand, *read it again*: if you *still* don’t understand it, *read it again*: if you fail, even after *three* readings, very likely your brain is getting a little tired. In that case, put the book away, and take to other occupations, and next day, when you come to it fresh, you will very likely find that it is *quite* easy.”

The same might be said for this volume, although readers are forgiven if they take a break for snacks after *two* readings.

about the author:

P.D. Magnus is a professor of philosophy in Albany, New York. His primary research is in the philosophy of science.