forall x

The Pitt Edition

An Introduction to Formal Logic

By **P. D. Magnus**
Tim Button
with additions and revisions by
J. Robert Loftis
Aaron Thomas-Bolduc
Richard Zach
J. Dmitri Gallow

This book is based on `forall`x: *Cambridge*, by

Tim Button
University of Cambridge

used under a CC BY 4.0 license, which is based in turn on `forall`x, by

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

used under a CC BY 4.0 license,
and was remixed, revised, & expanded by

Aaron Thomas-Bolduc & Richard Zach
University of Calgary

It includes additional material from `forall`x by P.D. Magnus and *Metatheory* by Tim Button, used under a CC BY 4.0 license, and from `forall`x: *Lorain County Remix*, by Cathal Woods and J. Robert Loftis. Further additions and revisions were made by J. Dmitri Gallow.

Contents

Preface

As the title indicates, this is a textbook on formal logic. Formal logic concerns the study of a certain kind of language which, like any language, can serve to express states of affairs. It is a formal language, i.e., its expressions (such as sentences) are defined formally. This makes it a very useful language for being very precise about the states of affairs its sentences describe. In particular, in formal logic it is impossible to be ambiguous. The study of these languages centres on the relationship of entailment between sentences, i.e., which sentences follow from which other sentences. Entailment is central because by understanding it better we can tell when some states of affairs must obtain provided some other states of affairs obtain. But entailment is not the only important notion. We will also consider the relationship of being satisfiable, i.e., of not being mutually contradictory. These notions can be defined semantically, using precise definitions of entailment based on interpretations of the language—or proof-theoretically, using formal systems of deduction.

Formal logic is of course a central sub-discipline of philosophy, where the logical relationship of assumptions to conclusions reached from them is important. Philosophers investigate the consequences of definitions and assumptions and evaluate these definitions and assumptions on the basis of their consequences. It is also important in mathematics and computer science. In mathematics, formal languages are used to describe not "everyday" states of affairs, but mathematical states of affairs. Mathematicians are also interested in the consequences of definitions and assumptions, and for them it is equally important to establish these consequences (which they call "theorems") using completely precise and rigorous methods. Formal logic provides such methods. In computer science, formal logic is applied to describe the state and behaviours of computational systems, e.g., circuits, programs, databases, etc. Methods of formal logic can likewise be used to establish consequences of such descriptions, such as whether a circuit is error-free, whether a program does what it's intended to do, whether a database is consistent or if something is true of the data in it.

The book is divided into seven parts. Part I introduces the topic and notions of logic in an informal way, without introducing a formal language yet. Parts II–IV concern the formal language of sentence logic. In it, sentences are formed from basic sentences using a number of operators ('or', 'and', 'not', 'if …then') which just combine sentences into more complicated ones. We discuss logical notions such as entailment in two ways: semantically, using the method of truth tables (in Part III) and proof-theoretically, using a system of formal derivations (in Part IV). PartsV–VII deal with a more complicated language, that of predicate logic. It includes, in addition to the connectives of sentence logic, also names, predicates, and the so-called quantifiers. These additional elements of the language make it much more expressive than sentence logic, and we'll spend a fair amount of time investigating just how much one can express in it. Again, logical notions for the language of predicate logic are defined semantically, using interpretations, and proof-theoretically, using a more complex version of the formal derivation system introduced in Part IV.

In the appendices you'll find a discussion of alternative notations for the languages we discuss in this text, of alternative derivation systems, and a quick reference listing most of the important rules and definitions.

PART I

Key notions of logic

1 Arguments

Logic is the business of evaluating arguments; sorting the good from the bad.

In everyday language, we sometimes use the word 'argument' to talk about belligerent shouting matches. If you and a friend have an argument in this sense, things are not going well between the two of you. Logic is not concerned with such teeth-gnashing and hair-pulling. They are not arguments, in our sense; they are just disagreements.

An argument, as we will understand it, is something more like this:

> It is raining heavily.
> If you do not take an umbrella, you will get soaked.
> ∴ You should take an umbrella.

We here have a series of sentences. The three dots on the third line of the argument are read 'therefore.' They indicate that the final sentence expresses the *conclusion* of the argument. The two sentences before that are the *premises* of the argument. If you believe the premises, then the argument (perhaps) provides you with a reason to believe the conclusion.

This is the sort of thing that logicians are interested in. We will say that an argument is any collection of premises, together with a conclusion.

This Part 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.

In the example just given, we used individual sentences to express both of the argument's premises, and we used a third sentence to express the argument's conclusion. Many arguments are expressed in this way, but a single sentence can contain a complete argument. Consider:

> I was wearing my sunglasses; so it must have been sunny.

This argument has one premise followed by a conclusion.

Many arguments start with premises, and end with a conclusion, but not all of them. The argument with which this section began might equally have been presented with the conclusion at the beginning, like so:

> You should take an umbrella. After all, it is raining heavily. And if you do not take an umbrella, you will get soaked.

Equally, it might have been presented with the conclusion in the middle:

> It is raining heavily. Accordingly, you should take an umbrella, given that if you do not take an umbrella, you will get soaked.

When approaching an argument, we want to know whether or not the conclusion follows from the premises. So the first thing to do is to separate out the conclusion from the premises. As a guide, these words are often used to indicate an argument's conclusion:

so, therefore, hence, thus, accordingly, consequently

By contrast, these expressions often indicate that we are dealing with a premise, rather than a conclusion:

since, because, given that

But in analysing an argument, there is no substitute for a good nose.

1.1 Sentences

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.

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 tasty.'

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 (in our sense). '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.

Practice exercises

At the end of some chapters, there are exercises that review and explore the material covered in the chapter. There is no substitute for actually working through some problems, because learning logic is more about developing a way of thinking than it is about memorizing facts.

So here's the first exercise. Highlight the phrase which expresses the conclusion of each of these arguments:

1. It is sunny. So I should take my sunglasses.
2. It must have been sunny. I did wear my sunglasses, after all.
3. No one but you has had their hands in the cookie-jar. And the scene of the crime is littered with cookie-crumbs. You're the culprit!
4. Miss Scarlett and Professor Plum were in the study at the time of the murder. Reverend Green had the candlestick in the ballroom, and we know that there is no blood on his hands. Hence Colonel Mustard did it in the kitchen with the lead-piping. Recall, after all, that the gun had not been fired.

2 Valid arguments

In §1, we gave a very permissive account of what an argument is. To see just how permissive it is, consider the following:

There is a bassoon-playing dragon in the *Cathedra Romana*.
∴ Salvador Dali was a poker player.

We have been given a premise and a conclusion. So we have an argument. Admittedly, it is a *terrible* argument, but it is still an argument.

2.1 Two ways that arguments can go wrong

It is worth pausing to ask what makes the argument so weak. In fact, there are two sources of weakness. First: the argument's (only) premise is obviously false. The Pope's throne is only ever occupied by a hat-wearing man. Second: the conclusion does not follow from the premise of the argument. Even if there were a bassoon-playing dragon in the Pope's throne, we would not be able to draw any conclusion about Dali's predilection for poker.

What about the main argument discussed in §1? The premises of this argument might well be false. It might be sunny outside; or it might be that you can avoid getting soaked without taking an umbrella. But even if both premises were true, it does not necessarily show you that you should take an umbrella. Perhaps you enjoy walking in the rain, and you would like to get soaked. So, even if both premises were true, the conclusion might nonetheless be false.

The general point is that, for any argument, there are two ways that it might go wrong:

- One or more of the premises might be false.
- The conclusion might not follow from the premises.

It is often important to determine whether or not the premises of an argument are true. However, that is normally a task best left to experts in the field: as it might be historians, scientists, or whomever. In our role as *logicians*, we are more concerned with arguments *in general*. So we are (usually) more concerned with the second way in which arguments can go wrong.

So: we are interested in whether or not a conclusion *follows from* some premises. Don't, though, say that the premises *infer* the conclusion. Entailment is a relation between premises and conclusions; inference is something we do. So if you want to mention inference when the conclusion follows from the premises, you could say that *one may infer* the conclusion from the premises.

2.2 Validity

As logicians, we want to be able to determine when the conclusion of an argument follows from the premises. One way to put this is as follows. We want to know whether, if all the premises were true, the conclusion would also have to be true. This motivates a definition:

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

And we will say that an argument is *invalid* iff it is not valid.

An argument is INVALID if and only if it is possible for all of 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 while 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 from the premises. *If* both premises are true, *then* the conclusion just has to be true. So the argument is valid.

This highlights that valid arguments do not need to have true premises or true conclusions. 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, but the argument is invalid. If Paris were to declare independence from the rest of France, then the conclusion would be false, even though both of the premises would remain true. Thus, it is *possible* for the premises of this argument to be true and the conclusion false. So 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. It is about whether it is *possible* for all the premises to be true and the conclusion false. Nonetheless, we will say that an argument is SOUND if and only if it is both valid and all of its premises are true.

2.3 Inductive arguments

Many good arguments are invalid. Consider this one:

In January 1997, it rained in London.
In January 1998, it rained in London.
In January 1999, it rained in London.
In January 2000, it rained in London.
∴ It rains every January in London.

This argument generalises from observations about several cases to a conclusion about all cases. Such arguments are called INDUCTIVE arguments. The argument could be made stronger by adding additional premises before drawing the conclusion: In January 2001, it rained in London; In January 2002.... But, however many premises of this form we add, the argument will remain invalid. Even if it has rained in London in every January thus far, it remains *possible* that London will stay dry next January.

The point of all this is that inductive arguments—even good inductive arguments—are not valid. They are not *watertight*. Unlikely though it might be, it is *possible* for their conclusion to be false, even when all of their premises are true. In this book, we will set aside (entirely) the question of what makes for a good inductive argument. Our interest is simply in sorting the valid arguments from the invalid ones.

2.4 Proving Invalidity

Suppose you wanted to persuade someone that an argument was invalid. How would you do that? Well, what it is for an argument to be invalid is for it to be *possible* that all of the argument's premises are true, while its conclusion is simultaneously false. So, if you wanted to show that the argument is invalid, you could simply describe that possibility in enough detail to persuade a reasonable person that the premises of an argument could be true while the conclusion is false.

Consider the following arguments:

The earth moves around the sun.
∴ The sun does not move.

Raising the minimum wage reduces employment.
Bernie wants to raise the minimum wage.
∴ Bernie wants to reduce employment.

We have not discovered life on other planets.
∴ There is no life on other planets.

Each of these arguments is invalid. And we may demonstrate that they are invalid by describing possibilities in which their premises are true while their conclusions are false. For the first argument, consider the following possibility: the earth rotates around the sun, and the sun itself rotates around the black hole at the center of the milky way galaxy. This is possible (in fact, it's actual), and, in this possibility, the premise of the first argument is true, yet the conclusion is false. So, the argument is invalid.

For the second argument, consider the following possibility: raising the minimum wage does reduce employment; however, Bernie does not know this. He incorrectly believes that raising the minimum wage wouldn't affect employment. Bernie wants to raise the minimum wage, but does not want to reduce employment. Since this is possible (though perhaps not actual), the argument is invalid.

It's important to appreciate that, when it comes to figuring out whether the argument is valid or invalid, it simply doesn't matter whether the premises or the conclusion are *in fact* true or false. Perhaps raising the minimum wage reduces employment; perhaps not. Perhaps Bernie wants to lower employment; perhaps not. When it comes to figuring out whether the argument is valid, these questions are simply *irrelevant*.

For the third argument, consider the following possibility: life on other planets is hidden somewhere we would be unlikely to have yet found it. Though we have not yet found it, it is still out there. In this possibility, the premise of the argument is true, yet its conclusion is false. Since this is possible (though perhaps not actual), the argument is invalid.

Practice exercises

A. Which of the following arguments are valid? Which are invalid?

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

Abe Lincoln was either born in Illinois or he was once president.
Abe Lincoln was never president.
∴ Abe Lincoln was born in Illinois.

If I pull the trigger, Abe Lincoln will die.
I do not pull the trigger.
∴ Abe Lincoln will not die.

Abe Lincoln was either from France or from Luxemborg.
Abe Lincoln was not from Luxemborg.
∴ Abe Lincoln was from France.

If the world were to end today, then I would not need to get up tomorrow morning.
I will need to get up tomorrow morning.
∴ The world will not end today.

Joe is now 19 years old.
Joe is now 87 years old.
∴ Bob is now 20 years old.

B. Could there be:

1. A valid argument that has one false premise and one true premise?
2. A valid argument that has only false premises?
3. A valid argument with only false premises and a false conclusion?
4. An invalid argument that can be made valid by the addition of a new premise?
5. A valid argument that can be made invalid by the addition of a new premise?

In each case: if so, give an example; if not, explain why not.

C. For each of the following arguments, if it is invalid, say which of the possibilities show that it is invalid (there may be more than one).

1. Sammy loves everyone who owns a horse.
   Bobby doesn't own a horse.
   ∴ Sammy doesn't love Bobby.

   a. The argument is valid.
   b. Cameron is the only person who owns a horse, and Sammy loves Cameron. Bobby doesn't own a horse, and Sammy doesn't love him.
   c. Cameron is the only person who owns a horse, but Sammy doesn't love Cameron. Bobby doesn't own a horse, and Sammy doesn't love him.
   d. Cameron is the only person who owns a horse, but Sammy doesn't love Cameron. Bobby doesn't own a horse, and Sammy loves him.
   e. Cameron is the only person who owns a horse, and Sammy loves Cameron. Bobby doesn't own a horse, but Sammy still loves him.

2. Margaret Thatcher doesn't live in New York City.
   New York City is located in New York State.
   ∴ Margaret Thatcher doesn't live in New York State.

   a. The argument is valid.
   b. Margaret Thatcher is dead, so she doesn't live anywhere any longer.
   c. Margaret Thatcher lives in London, which is not in New York State.
   d. Margaret Thatcher lives in Buffalo, which is in New York State, but not in New York City.
   e. Margaret Thatcher lives in Manhattan, which is in both New York City and New York State.

3. All humans are mammals.
   Some mammals have hair.

∴ All humans have hair.

a. The argument is valid. Therefore, there is no possibility in which the premises are true and the conclusion false.
b. All humans are mammals and all whales are mammals. Whales have hair, but humans do not.
c. Humans are not mammals, and even though all mammals have hair, humans do not have hair.
d. Who's to say that all humans are mammals? Some humans believe that they are immortal spirits.
e. Humans are mammals, but they are hairless; so too is every other mammal—no mammals have any hair.

3 Other logical notions

In §2, we introduced the idea of a valid argument. This is one of the most important ideas in logic. In this section, we will introduce are some similarly important ideas.

3.1 Joint possibility

Consider these two sentences:

B1. Jane's only brother is shorter than her.
B2. Jane's only brother is taller than her.

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. Similarly, if B2 is true, then B1 must be false. It is impossible that both sentences are true together. These sentences are jointly impossible; they cannot all be true at the same time. This motivates the following definitions:

Sentences are JOINTLY POSSIBLE if and only if it is possible for them all to be true together.

Sentences are JOINTLY IMPOSSIBLE if and only if it is *im*possible for them all to be true together.

B1 and B2 are *jointly impossible*.

We can ask about the joint possibility of any number of sentences. For example, consider the following four 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 entail 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 sentences G1–G4 are jointly impossible. They cannot all be true together. (Note that the sentences G1, G3 and G4 are jointly impossible. But if sentences are already jointly impossible, adding an extra sentence to the mix will not make them jointly possible!)

3.2 Necessary truths, necessary falsehoods, and contingencies

In assessing arguments for validity, we care about what would be true *if* the premises were true, but some sentences just *must* be true. Consider these sentences:

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

In order to know if sentence 1 is true, you would need to look outside or check the weather channel. It might be true; it might be false. A sentence which is capable of being true and capable of being false (in different circumstances, of course) is called CONTINGENT.

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 it is not. That is a NECESSARY TRUTH.

Equally, you do not need to check the weather to determine whether or not sentence 3 is true. 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 finish this sentence; but it is impossible for it to be both raining and not raining in the same place and at the same time. So, whatever the world is like, it is not both raining here and not raining here. It is a NECESSARY FALSEHOOD.

Necessary 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 NECESSARILY EQUIVALENT.

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 NECESSARY TRUTH is a sentence that must be true, that could not possibly be false.

▹ A NECESSARY FALSEHOOD is a sentence that must be false, that could not possibly be true.

▹ A CONTINGENT SENTENCE is neither a necessary truth nor a necessary falsehood. It may be true but could have been false, or vice versa.

▹ Two sentences are NECESSARILY EQUIVALENT if they must have the same truth value.

▹ A collection of sentences are JOINTLY POSSIBLE if it is possible for all these sentences to be true together; it is JOINTLY IMPOSSIBLE otherwise.

Practice exercises

A. For each of the following: Is it a necessary truth, a necessary falsehood, or contingent?

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.

B. For each of the following: Is it a necessary truth, a necessary falsehood, or contingent?

1. Elephants dissolve in water.
2. Wood is a light, durable substance useful for building things.
3. If wood were a good building material, it would be useful for building things.
4. I live in a three story building that is two stories tall.
5. If gerbils were mammals they would nurse their young.

C. Which of the following pairs of sentences are necessarily equivalent?

1. Elephants dissolve in water.
   If you put an elephant in water, it will disintegrate.
2. All mammals dissolve in water.
   If you put an elephant in water, it will disintegrate.
3. George Bush was the 43rd president.
   Barack Obama is the 44th president.
4. Barack Obama is the 44th president.
   Barack Obama was president immediately after the 43rd president.

5. Elephants dissolve in water.
   All mammals dissolve in water.

D. Which of the following pairs of sentences are necessarily equivalent?

1. Thelonious Monk played piano.
   John Coltrane played tenor sax.
2. Thelonious Monk played gigs with John Coltrane.
   John Coltrane played gigs with Thelonious Monk.
3. All professional piano players have big hands.
   Piano player Bud Powell had big hands.
4. Bud Powell suffered from severe mental illness.
   All piano players suffer from severe mental illness.
5. John Coltrane was deeply religious.
   John Coltrane viewed music as an expression of spirituality.

E. Consider the following 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.

Now consider each of the following collections of sentences. Which are jointly possible? Which are jointly impossible?

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

F. Consider the following sentences.

M1 All people are mortal.

M2 Socrates is a person.

M3 Socrates will never die.

M4 Socrates is mortal.

Which combinations of sentences are jointly possible? Mark each "possible" or "impossible."

1. Sentences M1, M2, and M3
2. Sentences M2, M3, and M4

3. Sentences M2 and M3
4. Sentences M1 and M4
5. Sentences M1, M2, M3, and M4

G. 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 necessary falsehood
4. An invalid argument, the conclusion of which is a necessary truth
5. A necessary truth that is contingent
6. Two necessarily equivalent sentences, both of which are necessary truths
7. Two necessarily equivalent sentences, one of which is a necessary truth and one of which is contingent
8. Two necessarily equivalent sentences that together are jointly impossible
9. A jointly possible collection of sentences that contains a necessary falsehood
10. An jointly impossible collection of sentences that contains a necessary truth

H. Which of the following is possible? If it is possible, give an example. If it is not possible, explain why.

1. A valid argument, whose premises are all necessary truths, and whose conclusion is contingent
2. A valid argument with true premises and a false conclusion
3. A jointly possible collection of sentences that contains two sentences that are not necessarily equivalent
4. A jointly possible collection of sentences, all of which are contingent
5. A false necessary truth
6. A valid argument with false premises
7. A necessarily equivalent pair of sentences that are jointly impossible
8. A necessary truth that is also a necessary falsehood
9. A jointly possible collection of sentences that are all necessary falsehoods

PART II

Sentence Logic

4 | First steps to symbolization

4.1 Validity in virtue of form

Consider this argument:

It is raining outside.
If it is raining outside, then Jenny is miserable.
∴ Jenny is miserable.

and another argument:

Jenny is an anarcho-syndicalist.
If Jenny is an anarcho-syndicalist, then Dipan is an avid reader of Tolstoy.
∴ Dipan is an avid reader of Tolstoy.

Both arguments are valid, and there is a straightforward sense in which we can say that they share a common structure. We might express the structure thus:

A
If A, then C
∴ C

This looks like an excellent argument *structure*. Indeed, surely any argument with this *structure* will be valid, and this is not the only good argument structure. Consider an argument like:

Jenny is either happy or sad.
Jenny is not happy.
∴ Jenny is sad.

Again, this is a valid argument. The structure here is something like:

A or B
not-A
∴ B

A superb structure! Here is another example:

It's not the case that Jim both studied hard and acted in lots of plays.

Jim studied hard
∴ Jim did not act in lots of plays.

This valid argument has a structure which we might represent thus:

not-(A and B)
A
∴ not-B

These examples illustrate an important idea, which we might describe as *validity in virtue of form*. The validity of the arguments just considered has nothing very much to do with the meanings of English expressions like 'Jenny is miserable', 'Dipan is an avid reader of Tolstoy', or 'Jim acted in lots of plays'. If it has to do with meanings at all, it is with the meanings of phrases like 'and', 'or', 'not,' and 'if..., then...'.

In Parts II–IV, we are going to develop a formal language which allows us to symbolize many arguments in such a way as to show that they are valid in virtue of their form. That language will be *sentence logic*, or SL.

4.2 Validity for special reasons

There are plenty of arguments that are valid, but not for reasons relating to their form. Take an example:

Juanita is a vixen
∴ Juanita is a fox

It is impossible for the premise to be true and the conclusion false. So the argument is valid. However, the validity is not related to the form of the argument. Here is an invalid argument with the same form:

Juanita is a vixen
∴ Juanita is a cathedral

This might suggest that the validity of the first argument *is* keyed to the meaning of the words 'vixen' and 'fox'. But, whether or not that is right, it is not simply the *shape* of the argument that makes it valid. Equally, consider the argument:

The sculpture is green all over.
∴ The sculpture is not red all over.

Again, it seems impossible for the premise to be true and the conclusion false, for nothing can be both green all over and red all over. So the argument is valid, but here is an invalid argument with the same form:

The sculpture is green all over.
∴ The sculpture is not shiny all over.

The argument is invalid, since it is possible to be green all over and shiny all over. (One might paint their nails with an elegant shiny green varnish.) Plausibly, the validity of the first argument is keyed to the way that colours (or colour-words) interact, but, whether or not that is right, it is not simply the *shape* of the argument that makes it valid.

The important moral can be stated as follows. *At best, SL will help us to understand arguments that are valid due to their form.*

4.3 Atomic sentences

We started isolating the form of an argument, in §4.1, by replacing *subsentences* of sentences with individual letters. Thus in the first example of this section, 'it is raining outside' is a subsentence of 'If it is raining outside, then Jenny is miserable', and we replaced this subsentence with 'A'.

Our artificial language, SL, pursues this idea absolutely ruthlessly. We start with some *sentence letters*. These will be the basic building blocks out of which more complex sentences are built. We will use single uppercase letters as sentence letters of SL. There are only twenty-six letters of the alphabet, but there is no limit to the number of sentence letters that we might want to consider. By adding subscripts to letters, we obtain new sentence letters. So, here are five different sentence letters of SL:

$$A, P, P_1, P_2, A_{234}$$

We will use sentence letters to represent, or *symbolize*, certain English sentences. To do this, we provide a SYMBOLIZATION KEY, such as the following:

A:	It is raining outside
C:	Jenny is miserable

In doing this, we are not fixing this symbolization *once and for all*. We are just saying that, for the time being, we will think of the sentence letter of SL, 'A', as symbolizing the English sentence 'It is raining outside', and the sentence letter of SL, 'C', as symbolizing the English sentence 'Jenny is miserable'. Later, when we are dealing with different sentences or different arguments, we can provide a new symbolization key; as it might be:

A:	Jenny is an anarcho-syndicalist
C:	Dipan is an avid reader of Tolstoy

It is important to understand that whatever structure an English sentence might have is lost when it is symbolized by a sentence letter of SL. From the point of view of SL, a sentence letter is just a letter. It can be used to build more complex sentences, but it cannot be taken apart.

5 Logical Operators

In the previous chapter, we considered symbolizing fairly basic English sentences with sentence letters of SL. This leaves us wanting to deal with the English expressions 'and', 'or', 'not', and so forth. These are *logical operators*—they can be used to form new sentences out of old ones. In SL, we will make use of logical operators to build complex sentences from atomic components. There are five logical operators in SL. This table summarizes them, and they are explained throughout this section.

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

These are not the only English operators of interest. Others are, e.g., 'unless', 'neither … nor …', and 'because'. We will see that the first two can be expressed by the operators we will discuss, while the last cannot. 'Because', in contrast to the others, is not *truth functional*.

5.1 Negation

Consider how we might symbolize these sentences:

1. Mary is in Barcelona.
2. It is not the case that Mary is in Barcelona.
3. Mary is not in Barcelona.

In order to symbolize sentence 1, we will need a sentence letter. We might offer this symbolization key:

B: Mary is in Barcelona.

Since sentence 2 is obviously related to sentence 1, we will not want to symbolize it with a completely different sentence. Roughly, sentence 2 means something like 'It is not the case that B'. In order to symbolize this, we need a symbol for negation. We will use '$\neg$'. Now we can symbolize sentence 2 with '$\neg B$'.

Sentence 3 also contains the word 'not', and it is obviously equivalent to sentence 2. As such, we can also symbolize it with '$\neg B$'.

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

It will help to offer a few more examples:

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

Let us use the following representation key:

R: The widget is replaceable

Sentence 4 can now be symbolized by 'R'. Moving on to sentence 5: saying the widget is irreplaceable means that it is not the case that the widget is replaceable. So even though sentence 5 does not contain the word 'not', we will symbolize it as follows: '$\neg R$'.

Sentence 6 can be paraphrased as 'It is not the case that the widget is irreplaceable.' Which can again be paraphrased as 'It is not the case that it is not the case that the widget is replaceable'. So we might symbolize this English sentence with the SL sentence '$\neg\neg R$'.

But some care is needed when handling negations. Consider:

7. Jane is happy.
8. Jane is unhappy.

If we let the SL-sentence 'H' symbolize 'Jane is happy', then we can symbolize sentence 7 as 'H'. However, it would be a mistake to symbolize sentence 8 with '$\neg H$'. If Jane is unhappy, then she is not happy; but sentence 8 does not mean the same thing as 'It is not the case that Jane is happy'. Jane might be neither happy nor unhappy; she might be in a state of blank indifference. In order to symbolize sentence 8, then, we would need a new sentence letter of SL.

5.2 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 of SL to symbolize sentences 9 and 10; perhaps

A: Adam is athletic.

B: Barbara is athletic.

Sentence 9 can now be symbolized as 'A', and sentence 10 can be symbolized as 'B'. Sentence 11 roughly says 'A and B'. We need another symbol, to deal with 'and'. We will use '$\wedge$'. Thus we will symbolize it as '$(A \wedge B)$'. This operator is called CONJUNCTION. We also say that 'A' and 'B' are the two CONJUNCTS of the conjunction '$(A \wedge B)$'.

Notice that we make no attempt to symbolize the word 'also' in sentence 11. Words like 'both' and 'also' function to draw our attention to the fact that two things are being conjoined. Maybe they affect the emphasis of a sentence, but we will not (and cannot) symbolize such things in SL.

Some more examples will bring out this point:

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

Sentence 12 is obviously a conjunction. The sentence says two things (about Barbara). In English, it is permissible to refer to Barbara only once. It *might* be tempting to think that we need to symbolize sentence 12 with something along the lines of 'B and energetic'. This would be a mistake. Once we symbolize part of a sentence as 'B', any further structure is lost, as 'B' is a sentence letter of SL. Conversely, 'energetic' is not an English sentence at all. What we are aiming for is something like 'B and Barbara is energetic'. So we need to add another sentence letter to the symbolization key. Let 'E' symbolize 'Barbara is energetic'. Now the entire sentence can be symbolized as '$(B \wedge E)$'.

Sentence 13 says one thing about two different subjects. It says of both Barbara and Adam that they are athletic, even though in English we use the word 'athletic' only once. The sentence can be paraphrased as 'Barbara is athletic, and Adam is athletic'. We can symbolize this in SL as '$(B \wedge A)$', using the same symbolization key that we have been using.

Sentence 14 is slightly more complicated. The word 'although' sets up a contrast between the first part of the sentence and the second part. Nevertheless, the sentence tells us both that Barbara is energetic and that she is not athletic. In order to make each of the conjuncts a sentence letter, 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'. Now we can symbolize this with the SL sentence '$(E \wedge \neg B)$'. Note that we have lost all sorts of nuance in this symbolization. There is a distinct difference in tone between sentence 14 and 'Both Barbara is energetic and it is not the case that Barbara is athletic'. SL does not (and cannot) preserve these nuances.

Sentence 15 raises similar issues. There is a contrastive structure, but this is not something that SL can deal with. So we can paraphrase the sentence as '*Both* Adam is athletic, *and* Barbara is more athletic than Adam'. (Notice that we once again replace the pronoun 'him' with 'Adam'.) How should we deal with the second conjunct? We already have the sentence letter 'A', which is being used to symbolize 'Adam is athletic', and the sentence 'B' which is being used to symbolize 'Barbara is athletic'; but neither of these concerns their relative athleticity. So, to symbolize the entire sentence, we need a new sentence letter. Let the SL sentence 'R' symbolize the English sentence 'Barbara is more athletic than Adam'. Now we can symbolize sentence 15 by '$(A \wedge R)$'.

A sentence can be symbolized as $(\mathscr{A} \wedge \mathscr{B})$ if it can be paraphrased in English as 'Both…, and…', or as '…, but …', or as 'although …, …'.

You might be wondering why we put parentheses around the conjunctions. The reason for this is brought out by considering how negation might interact with conjunction. Consider:

16. It's not the case that you will get both soup and salad.
17. You will not get soup but you will get salad.

Sentence 16 can be paraphrased as 'It is not the case that: both you will get soup and you will get salad'. Using this symbolization key:

S_1: You will get soup.
S_2: You will get salad.

We would symbolize 'both you will get soup and you will get salad' as '$(S_1 \wedge S_2)$'. To symbolize sentence 16, then, we simply negate the whole sentence, thus: '$\neg(S_1 \wedge S_2)$'.

Sentence 17 is a conjunction: you *will not* get soup, and you *will* get salad. 'You will not get soup' is symbolized by '$\neg S_1$'. So to symbolize sentence 17 itself, we offer '$(\neg S_1 \wedge S_2)$'.

These English sentences are very different, and their symbolizations differ accordingly. In one of them, the entire conjunction is negated. In the other, just one conjunct is negated. Parentheses help us to keep track of things like the *scope* of the negation.

5.3 Disjunction

Consider these sentences:

18. Either Fatima will play videogames, or she will watch movies.
19. Either Fatima or Omar will play videogames.

For these sentences we can use this symbolization key:

F: Fatima will play videogames.
O: Omar will play videogames.
M: Fatima will watch movies.

However, we will again need to introduce a new symbol. Sentence 18 is symbolized by '$(F \vee M)$'. The operator is called DISJUNCTION. We also say that 'F' and 'M' are the DISJUNCTS of the disjunction '$(F \vee M)$'.

Sentence 19 is only slightly more complicated. There are two subjects, but the English sentence only gives the verb once. However, we can paraphrase sentence 19 as 'Either Fatima will play videogames, or Omar will play videogames'. Now we can obviously symbolize it by '$(F \vee O)$' again.

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

Sometimes in English, the word 'or' is used in a way that 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 19, above. Fatima might play videogames alone, Omar might play videogames alone, or they might both play. Sentence 19 merely says that *at least* one of them plays videogames. This is called an INCLUSIVE OR. The SL symbol '$\vee$' always symbolizes an *inclusive or*.

It might help to see negation interact with disjunction. Consider:

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

Using the same symbolization key as before, sentence 20 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'. To symbolize this in SL, we need both disjunction and negation. 'It is not the case that you get soup' is symbolized by '$\neg S_1$'. 'It is not the case that you get salad' is symbolized by '$\neg S_2$'. So sentence 20 itself is symbolized by '$(\neg S_1 \vee \neg S_2)$'.

Sentence 21 also requires negation. It can be paraphrased as, '*It is not the case that* either you get soup or you get salad'. Since this negates the entire disjunction, we symbolize sentence 21 with '$\neg(S_1 \vee S_2)$'.

Sentence 22 is an *exclusive or*. We can break the sentence into two parts. The first part says that you get one or the other. We symbolize 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 symbolize this with '$\neg(S_1 \wedge S_2)$'. Now we just need to put the two parts together. As we saw above, 'but' can usually be symbolized with '$\wedge$'. Sentence 22 can thus be symbolized as '$((S_1 \vee S_2) \wedge \neg(S_1 \wedge S_2))$'.

This last example shows something important. Although the SL symbol '$\vee$' always symbolizes *inclusive or*, we can symbolize an *exclusive or* in SL. We just have to use a few of our other symbols as well.

5.4 Conditional

Consider these sentences:

23. If Jean is in Paris, then Jean is in France.
24. Jean is in France only if Jean is in Paris.

Let's use the following symbolization key:

P: Jean is in Paris.
F: Jean is in France

Sentence 23 is roughly of this form: 'if P, then F'. We will use the symbol '$\rightarrow$' to symbolize this 'if…, then…' structure. So we symbolize sentence 23 by '$(P \rightarrow F)$'. The operator is called THE CONDITIONAL. Here, 'P' is called the ANTECEDENT of the conditional '$(P \rightarrow F)$', and 'F' is called the CONSEQUENT.

Sentence 24 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 23. That would be a mistake. Your knowledge of geography tells you that sentence 23 is unproblematically true: there is no way for Jean to be in Paris that doesn't involve Jean being in France. But sentence 24 is not so straightforward: were Jean in Dieppe, Lyons, or Toulouse, Jean would be in France without being in Paris, thereby rendering sentence 24 false. Since geography alone dictates the truth of sentence 23, whereas travel plans (say) are needed to know the truth of sentence 24, they must mean different things.

In fact, sentence 24 can be paraphrased as 'If Jean is in France, then Jean is in Paris'. So we can symbolize it by '$(F \rightarrow P)$'.

A sentence can be symbolized as $\mathscr{A} \rightarrow \mathscr{B}$ if it can be paraphrased in English as 'If A, then B' or 'A only if B'.

In fact, many English expressions can be represented using the conditional. Consider:

25. For Jean to be in Paris, it is necessary that Jean be in France.
26. It is a necessary condition on Jean's being in Paris that she be in France.
27. For Jean to be in France, it is sufficient that Jean be in Paris.
28. It is a sufficient condition on Jean's being in France that she be in Paris.

If we think deeply about it, all four of these sentences mean the same as 'If Jean is in Paris, then Jean is in France'. So they can all be symbolized by '$P \rightarrow F$'.

It is important to bear in mind that the operator '$\rightarrow$' tells us only that, if the antecedent is true, then the consequent is true. It says nothing about a *causal* connection between two events (for example). In fact, we lose a huge amount when we use '$\rightarrow$' to symbolize English conditionals. We will return to this in §§9.3 and 11.5.

5.5 Biconditional

Consider these sentences:

29. Laika is a dog only if she is a mammal
30. Laika is a dog if she is a mammal
31. Laika is a dog if and only if she is a mammal

We will use the following symbolization key:

D:	Laika is a dog
M:	Laika is a mammal

Sentence 29, for reasons discussed above, can be symbolized by '$D \rightarrow M$'.

Sentence 30 is importantly different. It can be paraphrased as, 'If Laika is a mammal then Laika is a dog'. So it can be symbolized by '$M \rightarrow D$'.

Sentence 31 says something stronger than either 29 or 30. It can be paraphrased as 'Laika is a dog if Laika is a mammal, and Laika is a dog only if Laika is a mammal'. This is just the conjunction of sentences 29 and 30. So we can symbolize it as '$(D \rightarrow M) \wedge (M \rightarrow D)$'. We call this a BICONDITIONAL, because it entails the conditional in both directions.

We could treat every biconditional this way. So, just as we do not need a new SL symbol to deal with *exclusive or*, we do not really need a new SL symbol to deal with biconditionals. Because the biconditional occurs so often, however, we will use the symbol '$\leftrightarrow$' for it. We can then symbolize sentence 31 with the SL sentence '$D \leftrightarrow M$'.

The expression 'if and only if' occurs a lot especially in philosophy, mathematics, and logic. For brevity, we can abbreviate it with the snappier word 'iff'. We will follow this practice. So 'if' with only *one* 'f' is the English conditional. But 'iff' with *two* 'f's is the English biconditional. Armed with this we can say:

A sentence can be symbolized as $\mathscr{A} \leftrightarrow \mathscr{B}$ if it can be paraphrased in English as 'A iff B'; that is, as 'A if and only if B'.

A word of caution. Ordinary speakers of English often use 'if …, then…' when they really mean to use something more like '…if and only if …'. Perhaps your parents told you, when you were a child: 'if you don't eat your greens, you won't get any dessert'. Suppose you ate your greens, but that your parents refused to give you any dessert, on the grounds that they were only committed to the *conditional* (roughly 'if you get dessert, then you will have eaten your greens'), rather than the biconditional (roughly, 'you get dessert iff you eat your greens'). Well, a tantrum would rightly ensue. So, be aware of this when interpreting people; but in your own writing, make sure you use the biconditional iff you mean to.

5.6 Unless

We have now introduced all of the operators of SL. We can use them together to symbolize many kinds of sentences. An especially difficult case is when we use the English-language operator 'unless':

32. Unless you wear a jacket, you will catch a cold.
33. You will catch a cold unless you wear a jacket.

These two sentences are clearly equivalent. To symbolize them, we will use the symbolization key:

J: You will wear a jacket.
D: You will catch a cold.

Both sentences mean that if you do not wear a jacket, then you will catch a cold. With this in mind, we might symbolize them as '$\neg J \rightarrow D$'.

Equally, both sentences mean that if you do not catch a cold, then you must have worn a jacket. With this in mind, we might symbolize them as '$\neg D \rightarrow J$'.

Equally, both sentences mean that either you will wear a jacket or you will catch a cold. With this in mind, we might symbolize them as '$J \lor D$'.

All three are correct symbolizations. Indeed, in chapter 11 we will see that all three symbolizations are equivalent in SL.

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

Again, though, there is a little complication. 'Unless' can be symbolized as a conditional; but as we said above, people often use the conditional (on its own) when they mean to use the biconditional. Equally, 'unless' can be symbolized as a disjunction; but there are two kinds of disjunction (exclusive and inclusive). So it will not surprise you to discover that ordinary speakers of English often use 'unless' to mean something more like the biconditional, or like exclusive disjunction. Suppose someone says: 'I will go running unless it rains'. They probably mean something like 'I will go running iff it does not rain' (i.e. the biconditional), or 'either I will go running or it will rain, but not both' (i.e. exclusive disjunction). Again: be aware of this when interpreting what other people have said, but be precise in your writing.

Practice exercises

A. Using the symbolization key given, symbolize each English sentence in 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.

B. Using the symbolization key given, symbolize each English sentence in 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!

C. Using the symbolization key given, symbolize each English sentence in SL.

E_1:	Ava is an electrician.
E_2:	Harrison is an electrician.
F_1:	Ava is a firefighter.
F_2:	Harrison is a firefighter.
S_1:	Ava is satisfied with her career.
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.

D. Using the symbolization key given, symbolize each English-language sentence in SL.

J_1:	John Coltrane played tenor sax.
J_2:	John Coltrane played soprano sax.
J_3:	John Coltrane played tuba
M_1:	Miles Davis played trumpet
M_2:	Miles Davis played tuba

1. John Coltrane played tenor and soprano sax.
2. Neither Miles Davis nor John Coltrane played tuba.
3. John Coltrane did not play both tenor sax and tuba.
4. John Coltrane did not play tenor sax unless he also played soprano sax.
5. John Coltrane did not play tuba, but Miles Davis did.
6. Miles Davis played trumpet only if he also played tuba.
7. If Miles Davis played trumpet, then John Coltrane played at least one of these three instruments: tenor sax, soprano sax, or tuba.
8. If John Coltrane played tuba then Miles Davis played neither trumpet nor tuba.
9. Miles Davis and John Coltrane both played tuba if and only if Coltrane did not play tenor sax and Miles Davis did not play trumpet.

E. Give a symbolization key and symbolize the following English 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.

F. Give a symbolization key and symbolize the following English sentences in SL.

1. If there is food to be found in the pridelands, then Rafiki will talk about squashed bananas.

2. Rafiki will talk about squashed bananas unless Simba is alive.
3. Rafiki will either talk about squashed bananas or he won't, but there is food to be found in the pridelands regardless.
4. Scar will remain as king if and only if there is food to be found in the pridelands.
5. If Simba is alive, then Scar will not remain as king.

G. For each argument, write a symbolization key and symbolize all of the sentences of the argument in 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 on Tuesday. If it rains, Neville will be sad. If it snows, Neville will be cold. Therefore, Neville will either be sad or cold on Tuesday.
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.

H. For each argument, write a symbolization key and symbolize the argument as well as possible in SL. The part of the passage in italics is there to provide context for the argument, and doesn't need to be symbolized.

1. It is going to rain soon. I know because my leg is hurting, and my leg hurts if it's going to rain.
2. *Spider-man tries to figure out the bad guy's plan.* If Doctor Octopus gets the uranium, he will blackmail the city. I am certain of this because if Doctor Octopus gets the uranium, he can make a dirty bomb, and if he can make a dirty bomb, he will blackmail the city.
3. *A westerner tries to predict the policies of the Chinese government.* If the Chinese government cannot solve the water shortages in Beijing, they will have to move the capital. They don't want to move the capital. Therefore they must solve the water shortage. But the only way to solve the water shortage is to divert almost all the water from the Yangzi river northward. Therefore the Chinese government will go with the project to divert water from the south to the north.

I. We symbolized an *exclusive or* using '$\vee$', '$\wedge$', and '$\neg$'. How could you symbolize an *exclusive or* using only two operators? Is there any way to symbolize an *exclusive or* using only one operator?

6 | Sentences of SL

In general, we can specify a language by doing three things: 1) giving the vocabulary for the language, 2) giving the grammar of the language—that is, specifying which ways of sticking together the expressions from the vocabulary count as grammatical *sentences*, and 3) saying what the *meaning* of every grammatical expression is. For instance, in English, the vocabulary consists of all of the words of English. The grammar for English consists of rules saying when various strings of English words count as grammatical English sentences. 'Bubbie makes pickles' and 'Colorless green ideas sleep furiously' will count as sentences, whereas 'Up bouncy ball door John variously catapult' does not count as a sentence. Finally, the meaning of every English sentence is given by providing a dictionary entry for every word of English and providing rules for understanding the meaning of sentences in terms of the meanings of the words appearing in the sentence. The first two tasks are the tasks of specifying the *syntax* of the language. The final task is the fast of specifying the *semantics* of the language.

$$\begin{array}{l} \text{SYNTAX} \;—\left\{ \begin{array}{l} \text{1. Vocabulary} \\ \text{2. Grammar} \end{array} \right. \\ \text{SEMANTICS} \;—\text{3. Meaning} \end{array}$$

That's exactly what we're going to do for our artificial language SL. However, our task will be much simpler than the task of specifying English, as we will have a far simpler vocabulary, a far simpler grammar, and a far simpler semantics.

In this chapter, we will provide a *syntax* for SL. In chapters 8–10 we will build up SL's *semantics*.

6.1 Vocabulary

The vocabulary of SL includes the following symbols:

Atomic sentences	$A, B, C, \ldots, Z$
with subscripts, as needed	$A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$
Operators	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
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 of SL.

6.2 Sentences

Of course, many expressions of SL will be total gibberish. For instance, all of the following are expressions of SL:

$$\begin{array}{l} ((()A_{23} \land \land \to\to Z \\ P \to (Q) \to \land()) \\ (P \to (Q \to (R \to (S \to T)))) \\ A \land B \land (C\neg D))) \end{array}$$

However, only one—the third—is a *sentence* of SL. We want to know: when does an expression of SL amount to a sentence?

Obviously, individual sentence letters like 'A' and 'G_{13}' should count as sentences. (We'll also call them *atomic* sentences.) We can form further sentences out of these by using the various operators. Using negation, we can get '$\neg A$' and '$\neg G_{13}$'. Using conjunction, we can get '$(A \land G_{13})$', '$(G_{13} \land A)$', '$(A \land A)$', and '$(G_{13} \land G_{13})$'. We could also apply negation repeatedly to get sentences like '$\neg\neg A$' or apply negation along with conjunction to get sentences like '$\neg(A \land G_{13})$' and '$\neg(G_{13} \land \neg G_{13})$'. 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 sentences one by one.

Instead, we will describe the process by which sentences can be *constructed*. Consider negation: Given any sentence $\mathscr{A}$ of SL, $\neg\mathscr{A}$ is a sentence of SL. (Why the funny fonts? We return to this in §7.3.)

We can say similar things for each of the other operators. For instance, if $\mathscr{A}$ and $\mathscr{B}$ are sentences of SL, then $(\mathscr{A} \land \mathscr{B})$ is a sentence of SL. Providing clauses like this for all of the operators, we arrive at the following formal definition for a SENTENCE OF SL:

$\mathscr{A}$) Every sentence letter is a sentence.

$\neg$) If $\mathscr{A}$ is a sentence, then $\neg\mathscr{A}$ is a sentence.

$\land$) If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \land \mathscr{B})$ is a sentence.

$\lor$) If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \lor \mathscr{B})$ is a sentence.

$\to$) If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \to \mathscr{B})$ is a sentence.

$\leftrightarrow$) If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \leftrightarrow \mathscr{B})$ is a sentence.

$-$) Nothing else is a sentence.

Definitions like this are called *recursive*. Recursive definitions begin with some specifiable base elements, and then present ways to generate indefinitely many more elements by compounding together previously established ones. To give you a better idea of what a recursive definition is, we can give a recursive definition of the idea of *an ancestor of mine*. We specify a base clause.

- My parents are ancestors of mine.

and then offer further clauses like:

- If x is an ancestor of mine, then x's parents are ancestors of mine.
- Nothing else is an ancestor of mine.

Using this definition, we can easily check to see whether someone is my ancestor: just check whether she is the parent of the parent of…one of my parents. And the same is true for our recursive definition of sentences of SL. Just as the recursive definition allows complex sentences to be built up from simpler parts, the definition allows us to decompose sentences into their simpler parts. Once we get down to sentence letters, then we know we are ok.

Let's consider some examples.

Suppose we want to know whether or not '$\neg\neg\neg D$' is a sentence of SL. Looking at the second clause of the definition, we know that '$\neg\neg\neg D$' is a sentence *if* '$\neg\neg D$' is a sentence. So now we need to ask whether or not '$\neg\neg D$' is a sentence. Again looking at the second clause of the definition, '$\neg\neg D$' is a sentence *if* '$\neg D$' is. So, '$\neg D$' is a sentence *if* 'D' is a sentence. Now 'D' is a sentence letter of SL, so we know that 'D' is a sentence by the first clause of the definition ($\mathscr{A}$). So for a compound sentence like '$\neg\neg\neg D$', we must apply the definition repeatedly. Eventually we arrive at the sentence letters from which the sentence is built up.

Alternatively, to show that '$\neg\neg\neg D$' is a sentence of SL, we could show how it could be built up from the rules for sentences, as follows.

a) 'D' is a sentence [from ($\mathscr{A}$)]

b) So, '$\neg D$' is a sentence [from (a) and ($\neg$)]

c) So, '$\neg\neg D$' is a sentence [from (b) and ($\neg$)]

d) So, '$\neg\neg\neg D$' is a sentence [from (c) and ($\neg$)]

That is, we may build up to a longer, more complicated sentence by first establishing that some of its sub-expressions are sentences, and then using this to establish that the expression itself is a sentence.

Next, consider the example '$\neg(P \wedge \neg(\neg Q \vee R))$'. Looking at the second clause of the definition, ($\neg$), this is a sentence if '$(P \wedge \neg(\neg Q \vee R))$' is, and this is a sentence if *both* 'P' *and* '$\neg(\neg Q \vee R)$' are sentences. The former is a sentence letter, and the latter is a sentence if '$(\neg Q \vee R)$' is a sentence. It is—looking at the fourth clause of the definition, this is a sentence if both ‘$\neg Q$’ and ‘R’ are sentences. ‘R’ is a sentence letter, and, since ‘Q’ is a sentence letter, ‘$\neg Q$’ is a sentence by clause ($\neg$).

Or, alternatively, we could persuade ourselves that ‘$\neg(P \wedge \neg(\neg Q \vee R))$’ is a sentence by building it up according to the rules:

a) ‘Q’ is a sentence	[from ($\mathscr{A}$)]
b) So, ‘$\neg Q$’ is a sentence	[from (a) and ($\neg$)]
c) ‘R’ is a sentence	[from ($\mathscr{A}$)]
d) So, ‘$(\neg Q \vee R)$’ is a sentence	[from (b), (c), and ($\vee$)]
e) So ‘$\neg(\neg Q \vee R)$’ is a sentence	[from (d) and ($\neg$)]
f) ‘P’ is a sentence	[from ($\mathscr{A}$)]
g) So, ‘$(P \wedge \neg(\neg Q \vee R))$’ is a sentence	[from (e), (f), and ($\wedge$)]
h) So, ‘$\neg(P \wedge \neg(\neg Q \vee R))$’ is a sentence	[from (g) and ($\neg$)]

Ultimately, every sentence is constructed nicely out of sentence letters. When we are dealing with a sentence other than a sentence letter, we can see that there must be some sentential operator that was introduced *last*, when constructing the sentence. We call that operator the MAIN OPERATOR of the sentence.

The MAIN OPERATOR of a sentence of SL is the operator which would be added *last*, when building the sentence up according to the rules for sentences.

In the case of ‘$\neg\neg\neg D$’, the main operator is the very first ‘$\neg$’ sign. In the case of ‘$(P \wedge \neg(\neg Q \vee R))$’, the main operator is ‘$\wedge$’. In the case of ‘$((\neg E \vee F) \rightarrow \neg\neg G)$’, the main operator is ‘$\rightarrow$’.

The sentence’s *main operator* is just the operator associated with the last rule which would have to be applied if we were building the formula up by applying the rules for sentences above. For instance, if we want to know what the main operator is for the sentence ‘$(\neg P \wedge Q)$’, we would just imagine running through the following proof that ‘$(\neg P \wedge Q)$’ is a sentence of SL, by applying to the rules for sentences, *i.e.*,

a) ‘P’ is a sentence	[from ($\mathscr{A}$)]
b) So, ‘$\neg P$’ is a sentence	[from (a) and ($\neg$)]
c) ‘Q’ is a sentence	[from ($\mathscr{A}$)]
d) So, ‘$(\neg P \wedge Q)$’ is a sentence	[from (b), (c), and ($\wedge$)]

Here, the fact that we had to appeal to the rule ($\wedge$) in the final step of building up ‘$\neg P \wedge Q$’ tells us that $\wedge$ is the main operator. Imagine that we had tried to build up the formula in some other way. For instance, suppose we had attempted to *first* apply the rule ($\wedge$) and *then* the rule ($\neg$). Then, our proof would have gone line this.

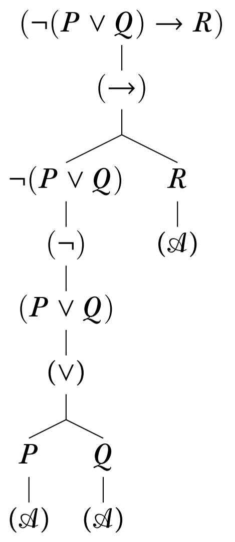
a) ‘P’ is a sentence	[from ($\mathscr{A}$)]
b) ‘Q’ is a sentence	[from ($\mathscr{A}$)]
c) So, ‘$(P \wedge Q)$’ is a sentence	[from (a), (b), and ($\wedge$)]
d) So, ‘$\neg(P \wedge Q)$’ is a sentence	[from (c) and ($\neg$)]

This is an entirely different sentence. '$\neg(P \wedge Q)$' is not the same as '$(\neg P \wedge Q)$'. While the main operator of '$(\neg P \wedge Q)$' is $\wedge$, the main operator of '$\neg(P \wedge Q)$' is $\neg$.

We can also use the rules for sentences to give a definition of what a sentence's *subsentences* are. $\mathscr{B}$ is a subsentence of $\mathscr{A}$ if and only if, in the course of building up $\mathscr{A}$ by applying the rules for sentences, $\mathscr{B}$ appears on a line before $\mathscr{A}$. So, for instance '$\neg P$' is a subsentence of '$(\neg P \wedge Q)$' (because it shows up on line (b) of our proof that '$(\neg P \wedge Q)$' is a sentence), whereas '$\neg P$' is *not* a subsentence of '$\neg(P \wedge Q)$' (since it does not show up at any point in our proof that '$\neg(P \wedge Q)$' is a sentence).

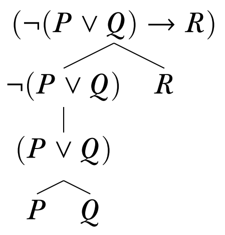
A sentences's *immediate subsentences* are those sentences whose lines would be appealed to in the final step of building the sentence up. For instance, the immediate subsentences of '$(\neg P \wedge Q)$' are '$\neg P$' and 'Q', whereas the immediate subsentence of '$\neg(P \wedge Q)$' is '$(P \wedge Q)$'. A sentence's immediate subsentences are just those sentences on which the sentence's main operator operates.

Another way of notating the proofs that certain expressions are sentences of SL is with SYNTAX TREES. For instance, we could represent a proof that '$(\neg(P \vee Q) \rightarrow R)$' is a sentence of SL with the following syntax tree.

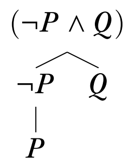


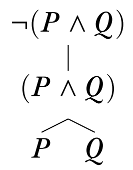
This tree tells us, firstly, that 'P' and 'Q' are sentences of PL (by rule $(\mathscr{A})$). Then, by rule $(\vee)$, '$(P \vee Q)$' is a sentence. Then, by rule $(\neg)$, '$\neg(P \vee Q)$' is a sentence. And, since 'R' is a sentence, by $(\mathscr{A})$, '$(\neg(P \vee Q) \rightarrow R)$' is a sentence (by rule $(\rightarrow)$).

If we want to leave out the rules, we can represent this syntax tree more simply as follows.



We can similarly write out the syntax trees for '$(\neg P \wedge Q)$' and '$\neg(P \wedge Q)$' like so.





These trees give us the syntactic structure of a sentence of SL. They highlight what the parentheses were already telling us about what the main operator of the sentence is, what its subformulae are, and how the various subformulae are interrelated (how the sentence is built up out of its subformulae). For instance, the tree on the left tells us that the immediate subformulae of '$(\neg P \wedge Q)$' are '$\neg P$' and 'Q'. And the tree on the right tells us that the immediate subformula of '$\neg(P \wedge Q)$' is '$(P \wedge Q)$'.

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. We will return to this point in Part III.

The recursive structure of sentences in SL also allows us to give a formal definition of the *scope* of a negation (mentioned in §5.2). The scope of a '$\neg$' is the subsentence for which '$\neg$' is the main logical operator. Consider a sentence like:

$$(P \wedge (\neg(R \wedge B) \leftrightarrow Q))$$

which was constructed by conjoining 'P' with '$(\neg(R \wedge B) \leftrightarrow Q)$'. This last sentence was constructed by placing a biconditional between '$\neg(R \wedge B)$' and 'Q'. The former of these sentences—a subsentence of our original sentence—is a sentence for which '$\neg$' is the main logical operator. So the scope of the negation is just '$\neg(R \wedge B)$'. More generally:

The SCOPE of an operator (in a sentence) is the sub-sentence for which that operator is the main operator.

6.3 Parenthesis conventions

Strictly speaking, the parentheses in '$(Q \wedge R)$' are an indispensable part of the sentence. Part of this is because we might use '$(Q \wedge R)$' as a subsentence in a more complicated sentence. For example, we might want to negate '$(Q \wedge R)$', obtaining '$\neg(Q \wedge R)$'. 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 '$(\neg Q \wedge R)$', but as we saw in §5.2, this is very different from '$\neg(Q \wedge R)$'.

Strictly speaking, then, '$Q \wedge R$' is *not* a sentence. It is a mere *expression*.

When working with SL, however, it will make our lives easier if we are sometimes a little less than strict. So, here are some convenient conventions.

First, we allow ourselves to omit the *outermost* parentheses of a sentence. Thus we allow ourselves to write '$Q \wedge R$' instead of the sentence '$(Q \wedge R)$'. However, we must remember to put the parentheses back in, when we want to embed the sentence into a more complicated sentence!

Second, it can be a bit painful to stare at long sentences with many nested pairs of parentheses. To make things a bit easier on the eyes, we will allow ourselves to use square parentheses, '[' and ']', instead of rounded ones. So there is no logical difference between '$(P \vee Q)$' and '$[P \vee Q]$', for example.

Combining these two conventions, we can rewrite the unwieldy sentence

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

rather more clearly as follows:

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

The scope of each operator is now much easier to pick out.

Practice exercises

A. For each of the following: (a) Is it a sentence of SL, strictly speaking? (b) Is it a sentence of SL, allowing for our relaxed parenthesis 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. $(A \rightarrow (A \wedge \neg F)) \vee (D \leftrightarrow E)$
7. $[(Z \leftrightarrow S) \rightarrow W] \wedge [J \vee X]$
8. $(F \leftrightarrow \neg D \rightarrow J) \vee (C \wedge D)$

B. Are there any sentences of SL that contain no sentence letters? Explain your answer.

C. What is the scope of each connective in the sentence

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

7 | Use and mention

In this Part, we have talked a lot *about* sentences. So we should pause to explain an important, and very general, point.

7.1 Quotation conventions

Consider these two sentences:

- Justin Trudeau is the Prime Minister.
- The expression 'Justin Trudeau' is composed of two uppercase letters and eleven lowercase letters

When we want to talk about the Prime Minister, we *use* his name. When we want to talk about the Prime Minister's name, we *mention* that name, which we do by putting it in quotation marks.

There is a general point here. When we want to talk about things in the world, we just *use* words. When we want to talk about words, we typically have to *mention* those words. We need to indicate that we are mentioning them, rather than using them. To do this, some convention is needed. We can put them in quotation marks, or display them centrally in the page (say). So this sentence:

- 'Justin Trudeau' is the Prime Minister.

says that some *expression* is the Prime Minister. That's false. The *man* is the Prime Minister; his *name* isn't. Conversely, this sentence:

- Justin Trudeau is composed of two uppercase letters and eleven lowercase letters.

also says something false: Justin Trudeau is a man, made of flesh rather than letters. One final example:

- " 'Justin Trudeau' " is the name of 'Justin Trudeau'.

On the left-hand-side, here, we have the name of a name. On the right hand side, we have a name. Perhaps this kind of sentence only occurs in logic textbooks, but it is true nonetheless.

Those are just general rules for quotation, and you should observe them carefully in all your work! To be clear, the quotation-marks here do not indicate indirect speech. They indicate that you are moving from talking about an object, to talking about the name of that object.

7.2 Object language and metalanguage

These general quotation conventions are of particular importance for us. After all, we are describing a formal language here, SL, and so we are often *mentioning* expressions from 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.

For the most part, the object language in this chapter has been the formal language that we have been developing: SL. The metalanguage is English. Not conversational English exactly, but English supplemented with some additional vocabulary which helps us to get along.

Now, we have used uppercase letters as sentence letters of SL:

$$A, B, C, Z, A_1, B_4, A_{25}, J_{375}, \ldots$$

These are sentences of the object language (SL). They are not sentences of English. So we must not say, for example:

- D is a sentence letter of SL.

Obviously, we are trying to come out with an English sentence that says something about the object language (SL), but 'D' is a sentence of SL, and not part of English. So the preceding is gibberish, just like:

- Schnee ist weiß is a German sentence.

What we surely meant to say, in this case, is:

- 'Schnee ist weiß' is a German sentence.

Equally, what we meant to say above is just:

- 'D' is a sentence letter of SL.

The general point is that, whenever we want to talk in English about some specific expression of SL, we need to indicate that we are *mentioning* the expression, rather than *using* it. We can either deploy quotation marks, or we can adopt some similar convention, such as placing it centrally in the page.

7.3 Metavariables

However, we do not just want to talk about *specific* expressions of SL. We also want to be able to talk about *any arbitrary* sentence of SL. Indeed, we had to do this in §6.2, when we presented the recursive definition of a sentence of SL. We used uppercase script letters to do this, namely:

$$\mathscr{A}, \mathscr{B}, \mathscr{C}, \mathscr{D}, \ldots$$

These symbols do not belong to SL. Rather, they are part of our (augmented) metalanguage that we use to talk about *any* expression of SL. To repeat the second clause of the recursive definition of a sentence of SL, we said:

2. If $\mathscr{A}$ is a sentence, then $\neg\mathscr{A}$ is a sentence.

This talks about *arbitrary* sentences. If we had instead offered:

- If 'A' is a sentence, then '$\neg A$' is a sentence.

this would not have allowed us to determine whether '$\neg B$' is a sentence. To emphasize, then:

'$\mathscr{A}$' is a symbol (called a METAVARIABLE) in augmented English, which we use to talk about any SL expression. 'A' is a particular sentence letter of SL.

But this last example raises a further complication for our quotation conventions. We have not included any quotation marks in the second clause of our recursive definition. Should we have done so?

The problem is that the expression on the right-hand-side of this rule is not a sentence of English, since it contains '$\neg$'. So we might try to write:

2′. If $\mathscr{A}$ is a sentence, then '$\neg\mathscr{A}$' is a sentence.

But this is no good: '$\neg\mathscr{A}$' is not a SL sentence, since '$\mathscr{A}$' is a symbol of (augmented) English rather than a symbol of SL.

What we really want to say is something like this:

2″. If $\mathscr{A}$ is a sentence, then the result of concatenating the symbol '$\neg$' with the sentence $\mathscr{A}$ is a sentence.

This is impeccable, but rather long-winded. But we can avoid long-windedness by creating our own conventions. We can perfectly well stipulate that an expression like '$\neg\mathscr{A}$' should simply be read *directly* in terms of rules for concatenation. So, *officially*, the metalanguage expression '$\neg\mathscr{A}$' simply abbreviates:

the result of concatenating the symbol '$\neg$' with the sentence $\mathscr{A}$

and similarly, for expressions like '$(\mathscr{A} \wedge \mathscr{B})$', '$(\mathscr{A} \vee \mathscr{B})$', etc.

7.4 Quotation conventions for arguments

One of our main purposes for using SL is to study arguments, and that will be our concern in Parts III and IV. In English, the premises of an argument are often expressed by individual sentences, and the conclusion by a further sentence. Since we can symbolize English sentences, we can symbolize English arguments using SL. Thus we might ask whether the argument whose premises are the SL sentences 'A' and '$A \rightarrow C$', and whose conclusion is the SL sentence 'C', is valid. However, it is quite a mouthful to write that every time. So instead we will introduce another bit of abbreviation. This:

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \therefore \mathscr{C}$$

abbreviates:

> the argument with premises $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ and conclusion $\mathscr{C}$

To avoid unnecessary clutter, we will not regard this as requiring quotation marks around it. (Note, then, that '$\therefore$' is a symbol of our augmented *metalanguage*, and not a new symbol of SL.)

PART III

Semantics for Sentence Logic

8 Characteristic truth tables

Any sentence of SL is composed of sentence letters, possibly combined using sentential operators. The truth value of the compound sentence depends only on the truth value of the sentence letters that comprise it. In order to know the truth value of '$(D \wedge E)$', for instance, you only need to know the truth value of 'D' and the truth value of 'E'.

We introduced five operators in chapter 5, so we simply need to explain how they map between truth values. For convenience, we will abbreviate 'True' with 'T' and 'False' with 'F'. (But just to be clear, the two truth values are True and False; the truth values are not *letters*!)

Negation. For any sentence $\mathscr{A}$: If $\mathscr{A}$ is true, then $\neg\mathscr{A}$ is false. If $\neg\mathscr{A}$ is true, then $\mathscr{A}$ is false. We can summarize this in the *characteristic truth table* for negation:

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

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

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

Note that conjunction is *symmetrical*. The truth value for $\mathscr{A} \wedge \mathscr{B}$ is always the same as the truth value for $\mathscr{B} \wedge \mathscr{A}$.

Disjunction. Recall that '$\vee$' always represents inclusive or. So, for any sentences $\mathscr{A}$ and $\mathscr{B}$, $\mathscr{A} \vee \mathscr{B}$ is true if and only if either $\mathscr{A}$ or $\mathscr{B}$ is true. We can summarize this in the characteristic truth table for disjunction:

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

Like conjunction, disjunction is symmetrical.

Conditional. We're just going to come clean and admit it: Conditionals are a right old mess in SL. Exactly how much of a mess they are is *philosophically* contentious. We'll discuss a few of the subtleties in §§9.3 and 11.5. For now, we are going to stipulate the following: $\mathscr{A} \rightarrow \mathscr{B}$ is false if and only if $\mathscr{A}$ is true and $\mathscr{B}$ is false. We can summarize this with a characteristic truth table for the conditional.

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{A} \rightarrow \mathscr{B}$
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 $\mathscr{A} \rightarrow \mathscr{B}$ has a very different truth table from $\mathscr{B} \rightarrow \mathscr{A}$.

Biconditional. Since a biconditional is to be the same as the conjunction of a conditional running in each direction, we will want the truth table for the biconditional to be:

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

Unsurprisingly, the biconditional is symmetrical.

9 | Truth-functional operators

9.1 The idea of truth-functionality

Let's introduce an important idea.

An operator is TRUTH-FUNCTIONAL iff the truth value of a sentence with that operator as its main operator is uniquely determined by the truth value(s) of the constituent subsentence(s).

Every operator in SL is truth-functional. The truth value of a negation is uniquely determined by the truth value of the unnegated sentence. The truth value of a conjunction is uniquely determined by the truth value of both conjuncts. The truth value of a disjunction is uniquely determined by the truth value of both disjuncts, and so on. To determine the truth value of some SL sentence, we only need to know the truth value of its components.

In plenty of languages there are operators that are not truth-functional. In English, for example, we can form a new sentence from any simpler sentence by prefixing it with 'It is necessarily the case that…'. The truth value of this new sentence is not fixed solely by the truth value of the original sentence. For consider two true sentences:

1. $2 + 2 = 4$
2. Shostakovich wrote fifteen string quartets

Whereas it is necessarily the case that $2 + 2 = 4$, it is not *necessarily* the case that Shostakovich wrote fifteen string quartets. If Shostakovich had died earlier, he would have failed to finish Quartet no. 15; if he had lived longer, he might have written a few more. So 'It is necessarily the case that…' is an operator of English, but it is not *truth-functional*.

Because the operators of SL are truth-functional, if we know the truth-value of all of the atomic sentences appearing in a complex sentence of SL, then we can use our knowledge of the syntactic structure of the complex sentence to determine *its* truth-value. For instance, suppose that we know that 'P' is true and 'Q' is false. Then, we know that '$\neg P \wedge Q$' is false and that '$\neg(P \wedge Q)$' is true: we simply plug in the truth-value T for 'P' and the truth-value F for 'Q' and work our way up the syntax tree:

$$\begin{tikzcd}
& (\neg P \wedge Q)[F] \arrow[dl, dash] \arrow[dr, dash] & \\
\neg P[F] \arrow[d, dash] & & Q[F] \\
P\,[T] & &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
& \neg(P \wedge Q)[T] \arrow[d, dash] & \\
& (P \wedge Q)[F] \arrow[dl, dash] \arrow[dr, dash] & \\
P[T] & & Q[F]
\end{tikzcd}$$

This is one reason why it is so important for us to be precise about the syntactic structure represented with the parentheses—it makes a difference with respect to whether a sentence of SL is true or false.

9.2 Symbolizing versus translating

All of the operators of SL are truth-functional, but more than that: they really do nothing *but* map us between truth values.

When we symbolize a sentence or an argument in SL, we ignore everything *besides* the contribution that the truth values of a component might make to the truth value of the whole. There are subtleties to our ordinary claims that far outstrip their mere truth values. Sarcasm; poetry; snide implicature; emphasis; these are important parts of everyday discourse, but none of this is retained in SL. As remarked in §5, SL cannot capture the subtle differences between the following English sentences:

1. Dana is a logician and Dana is a nice person
2. Although Dana is a logician, Dana is a nice person
3. Dana is a logician despite being a nice person
4. Dana is a nice person, but also a logician
5. Dana's being a logician notwithstanding, he is a nice person

All of the above sentences will be symbolized with the same SL sentence, perhaps '$L \wedge N$'.

We keep saying that we use SL sentences to *symbolize* English sentences. Many other textbooks talk about *translating* English sentences into SL. However, a good translation should preserve certain facets of meaning, and—as we have just pointed out—SL just cannot do that. This is why we will speak of *symbolizing* English sentences, rather than of *translating* them.

This affects how we should understand our symbolization keys. Consider a key like:

L: Dana is a logician.
N: Dana is a nice person.

Other textbooks will understand this as a stipulation that the SL sentence 'L' should *mean* that Dana is a logician, and that the SL sentence 'N' should *mean* that Dana is a nice person, but SL just is totally unequipped to deal with *meaning*. The preceding symbolization key is doing no more and no less than stipulating that the SL sentence

'*L*' should take the same truth value as the English sentence 'Dana is a logician' (whatever that might be), and that the SL sentence '*N*' should take the same truth value as the English sentence 'Dana is a nice person' (whatever that might be).

When we treat a SL sentence as *symbolizing* an English sentence, we are stipulating that the SL sentence is to take the same truth value as that English sentence.

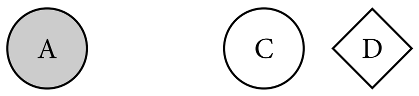
9.3 Indicative versus subjunctive conditionals

We want to bring home the point that SL can *only* deal with truth functions by considering the case of the conditional. When we introduced the characteristic truth table for the material conditional in §8, we did not say anything to justify it. Let's now offer a justification, which follows Dorothy Edgington.[1]

Suppose that Lara has drawn some shapes on a piece of paper, and coloured some of them in. We have not seen them, but nevertheless claim:

If any shape is grey, then that shape is also circular.

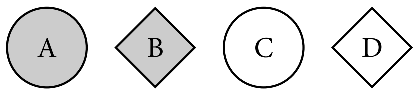
As it happens, Lara has drawn the following:



In this case, our claim is surely true. Shapes C and D are not grey, and so can hardly present *counterexamples* to our claim. Shape A *is* grey, but fortunately it is also circular. So my claim has no counterexamples. It must be true. That means that each of the following *instances* of our claim must be true too:

- If A is grey, then it is circular (true antecedent, true consequent)
- If C is grey, then it is circular (false antecedent, true consequent)
- If D is grey, then it is circular (false antecedent, false consequent)

However, if Lara had drawn a fourth shape, thus:



then our claim would be false. So it must be that this claim is false:

- If B is grey, then it is circular (true antecedent, false consequent)

[1] Dorothy Edgington, 'Conditionals', 2006, in the *Stanford Encyclopedia of Philosophy* (http://plato.stanford.edu/entries/conditionals/).

Now, recall that every operator of SL has to be truth-functional. This means that merely the truth values of the antecedent and consequent must uniquely determine the truth value of the conditional as a whole. Thus, from the truth values of our four claims—which provide us with all possible combinations of truth and falsity in antecedent and consequent—we can read off the truth table for the material conditional.

What this argument shows is that '→' is the *best* candidate for a truth-functional conditional. Otherwise put, *it is the best conditional that SL can provide*. But is it any good, as a surrogate for the conditionals we use in everyday language? Consider two sentences:

1. If Mitt Romney had won the 2012 election, then he would have been the 45th President of the USA.
2. If Mitt Romney had won the 2012 election, then he would have turned into a helium-filled balloon and floated away into the night sky.

Sentence 1 is true; sentence 2 is false, but both have false antecedents and false consequents. So the truth value of the whole sentence is not uniquely determined by the truth value of the parts. Do not just blithely assume that you can adequately symbolize an English 'if ..., then ...' with SL's '→'.

The crucial point is that sentences 1 and 2 employ *subjunctive* conditionals, rather than *indicative* conditionals. They ask us to imagine something contrary to fact—Mitt Romney lost the 2012 election—and then ask us to evaluate what *would* have happened in that case. Such considerations just cannot be tackled using '→'.

We will say more about the difficulties with conditionals in §11.5. For now, we will content ourselves with the observation that '→' is the only candidate for a truth-functional conditional for SL, but that many English conditionals cannot be represented adequately using '→'. SL is an intrinsically limited language.

10 | Complete truth tables

So far, we have considered assigning truth values to SL sentences indirectly. We have said, for example, that a SL sentence such as 'B' is to take the same truth value as the English sentence 'Big Ben is in London' (whatever that truth value may be), but we can also assign truth values *directly*. We can simply stipulate that 'B' is to be true, or stipulate that it is to be false.

A VALUATION is any assignment of truth values to particular sentence letters of SL.

The power of truth tables lies in the following. Each row of a truth table represents a possible valuation. The entire truth table represents all possible valuations; thus the truth table provides us with a means to calculate the truth values of complex sentences, on each possible valuation. This is easiest to explain by example.

10.1 A worked example

Consider the sentence '$(H \wedge I) \rightarrow H$'. There are four possible ways to assign True and False to the sentence letter 'H' and 'I'—four possible valuations—which we can represent as follows:

H	I	$(H \wedge I) \rightarrow H$
T	T	
T	F	
F	T	
F	F	

To calculate the truth value of the entire sentence '$(H \wedge I) \rightarrow H$', we first copy the truth values for the sentence letters and write them underneath the letters in the sentence:

H	I	$(H$	$\wedge$	$I)$	$\rightarrow$	H
T	T	T		T		T
T	F	T		F		T
F	T	F		T		F
F	F	F		F		F

Now consider the subsentence '$(H \land I)$'. This is a conjunction, $(\mathscr{A} \land \mathscr{B})$, with '$H$' as $\mathscr{A}$ and with 'I' as $\mathscr{B}$. The characteristic truth table for conjunction gives the truth conditions for *any* sentence of the form $(\mathscr{A} \land \mathscr{B})$, whatever $\mathscr{A}$ and $\mathscr{B}$ might be. It represents the point that a conjunction is true iff both conjuncts are true. In this case, our conjuncts are just 'H' and 'I'. They are both true on (and only on) the first line of the truth table. Accordingly, we can calculate the truth value of the conjunction on all four rows.

		$\mathscr{A}$	$\land$	$\mathscr{B}$		
H	I	$(H$	$\land$	$I)$	$\rightarrow$	H
T	T	T	T	T		T
T	F	T	F	F		T
F	T	F	F	T		F
F	F	F	F	F		F

Now, the entire sentence that we are dealing with is a conditional, $\mathscr{A} \rightarrow \mathscr{B}$, with '$(H \land I)$' as $\mathscr{A}$ and with 'H' as $\mathscr{B}$. On the second row, for example, '$(H \land I)$' is false and 'H' is true. Since a conditional is true when the antecedent is false, we write a 'T' in the second row underneath the conditional symbol. We continue for the other three rows and get this:

			$\mathscr{A}$		$\rightarrow$	$\mathscr{B}$
H	I	$(H$	$\land$	$I)$	$\rightarrow$	H
T	T		T		T	T
T	F		F		T	T
F	T		F		T	F
F	F		F		T	F

The conditional is the main logical operator of the sentence, so the column of 'T's underneath the conditional tells us that the sentence '$(H \land I) \rightarrow H$' 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. Since we have considered all four possible assignments of truth and falsity to 'H' and 'I'—since, that is, we have considered all the different *valuations*—we can say that '$(H \land I) \rightarrow H$' is true on every valuation.

In this example, we have not repeated all of the entries in every column 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	$(H$	$\land$	$I)$	$\rightarrow$	H
T	T	T	T	T	**T**	T
T	F	T	F	F	**T**	T
F	T	F	F	T	**T**	F
F	F	F	F	F	**T**	F

Most of the columns underneath the sentence are only there for bookkeeping purposes. The column that matters most is the column underneath the *main logical operator* for the sentence, since this tells you the truth value of the entire sentence. We have emphasized this, by putting this column in bold. When you work through truth tables yourself, you should similarly emphasize it (perhaps by underlining).

10.2 Building complete truth tables

A COMPLETE TRUTH TABLE has a line for every possible assignment of True and False to the relevant sentence letters. Each line represents a *valuation*, and a complete truth table has a line for all the different valuations.

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. The truth table for this sentence looks like this:

C	$[(C$	$\leftrightarrow$	$C)$	$\rightarrow$	$C]$	$\wedge$	$\neg$	$(C$	$\rightarrow$	$C)$
T	T	T	T	T	T	**F**	F	T	T	T
F	F	T	F	F	F	**F**	F	F	T	F

Looking at the column underneath the main logical operator, we see that the sentence is false on both rows of the table; i.e., the sentence is false regardless of whether 'C' is true or false. It is false on every valuation.

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

A sentence that contains three sentence letters requires eight lines:

M	N	P	M	$\wedge$	$(N$	$\vee$	$P)$
T	T	T	T	**T**	T	T	T
T	T	F	T	**T**	T	T	F
T	F	T	T	**T**	F	T	T
T	F	F	T	**F**	F	F	F
F	T	T	F	**F**	T	T	T
F	T	F	F	**F**	T	T	F
F	F	T	F	**F**	F	T	T
F	F	F	F	**F**	F	F	F

From this table, we know that the sentence '$M \wedge (N \vee P)$' can 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 lines.

In order to fill in the columns of a complete truth table, begin with the right-most sentence letter and alternate between 'T' and 'F'. In the next column to the left, write two 'T's, write two 'F's, and repeat. For the third sentence letter, write four 'T's followed by four 'F's. 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 'T's followed by eight 'F's. For a 32 line table, the next column would have 16 'T's followed by 16 'F's, and so on.

10.3 More about parentheses

Consider these two sentences:

$$((A \wedge B) \wedge C)$$
$$(A \wedge (B \wedge C))$$

These are truth functionally equivalent. Consequently, it will never make any difference from the perspective of truth value – which is all that SL cares about (see §9) – which of the two sentences we assert (or deny). Even though the order of the parentheses does not matter as to their truth, we should not just drop them. The expression

$$A \wedge B \wedge C$$

is ambiguous between the two sentences above. The same observation holds for disjunctions. The following sentences are logically equivalent:

$$((A \vee B) \vee C)$$
$$(A \vee (B \vee C))$$

But we should not simply write:

$$A \vee B \vee C$$

In fact, it is a specific fact about the characteristic truth table of $\vee$ and $\wedge$ that guarantees that any two conjunctions (or disjunctions) of the same sentences are truth functionally equivalent, however you place the parentheses. *But be careful.* These two sentences have *different* truth tables:

$$((A \rightarrow B) \rightarrow C)$$
$$(A \rightarrow (B \rightarrow C))$$

So if we were to write:

$$A \rightarrow B \rightarrow C$$

it would be dangerously ambiguous. So we must not do the same with conditionals. Equally, these sentences have different truth tables:

$$((A \vee B) \wedge C)$$
$$(A \vee (B \wedge C))$$

So if we were to write:

$$A \vee B \wedge C$$

it would be dangerously ambiguous. *Never write this.* The moral is: never drop parentheses. (Except, of course, for the outermost parentheses, which we allow ourselves to omit—omitting the outermost parentheses will never lead to this kind of ambiguity.)

Practice exercises

A. Offer complete truth tables for each of the following:

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

B. Check all the claims made in introducing the new notational conventions in §10.3, i.e. show that:

1. '$((A \wedge B) \wedge C)$' and '$(A \wedge (B \wedge C))$' have the same truth table
2. '$((A \vee B) \vee C)$' and '$(A \vee (B \vee C))$' have the same truth table
3. '$((A \vee B) \wedge C)$' and '$(A \vee (B \wedge C))$' do not have the same truth table
4. '$((A \to B) \to C)$' and '$(A \to (B \to C))$' do not have the same truth table

Also, check whether:

5. '$((A \leftrightarrow B) \leftrightarrow C)$' and '$(A \leftrightarrow (B \leftrightarrow C))$' have the same truth table

C. Write complete truth tables for the following sentences and mark the column that represents the possible truth values for the whole sentence.

1. $\neg(S \leftrightarrow (P \to S))$
2. $\neg[(X \wedge Y) \vee (X \vee Y)]$
3. $(A \to B) \leftrightarrow (\neg B \leftrightarrow \neg A)$
4. $[C \leftrightarrow (D \vee E)] \wedge \neg C$
5. $\neg(G \wedge (B \wedge H)) \leftrightarrow (G \vee (B \vee H))$

D. Write complete truth tables for the following sentences and mark the column that represents the possible truth values for the whole sentence.

1. $(D \wedge \neg D) \to G$

2. $(\neg P \vee \neg M) \leftrightarrow M$
3. $\neg\neg(\neg A \wedge \neg B)$
4. $[(D \wedge R) \rightarrow I] \rightarrow \neg(D \vee R)$
5. $\neg[(D \leftrightarrow O) \leftrightarrow A] \rightarrow (\neg D \wedge O)$

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

11 Semantic concepts

In the previous section, we introduced the idea of a valuation and showed how to determine the truth value of any SL sentence, on any valuation, using a truth table. In this section, we will introduce some related ideas, and show how to use truth tables to test whether or not they apply.

11.1 Tautologies and contradictions

In §3, we explained *necessary truth* and *necessary falsity*. Both notions have surrogates in SL. We will start with a surrogate for necessary truth.

$\mathscr{A}$ is a TAUTOLOGY (in SL) iff it is true on every valuation.

We can determine whether a sentence is a tautology (in SL) just by using truth tables. If the sentence is true on every line of a complete truth table, then it is true on every valuation, so it is a tautology. In the example of §10, '$(H \land I) \rightarrow H$' is a tautology.

This is only, though, a *surrogate* for necessary truth. There are some necessary truths that we cannot adequately symbolize in SL. An example is '$2 + 2 = 4$'. This *must* be true, but if we try to symbolize it in SL, the best we can offer is an sentence letter, and no sentence letter is a tautology. Still, if we can adequately symbolize some English sentence using an SL sentence which is a tautology, then that English sentence expresses a necessary truth.

We have a similar surrogate for necessary falsity:

$\mathscr{A}$ is a CONTRADICTION (in SL) iff it is false on every valuation.

We can determine whether a sentence is a contradiction just by using truth tables. If the sentence is false on every line of a complete truth table, then it is false on every valuation, so it is a contradiction. In the example of §10, '$[(C \leftrightarrow C) \rightarrow C] \land \neg(C \rightarrow C)$' is a contradiction.

11.2 Equivalence

Here is a similar useful notion:

> $\mathscr{A}$ and $\mathscr{B}$ are EQUIVALENT (in SL) iff, for every valuation, their truth values agree, i.e. if there is no valuation in which they have opposite truth values.

We have already made use of this notion, in effect, in §10.3; the point was that '$(A \wedge B) \wedge C$' and '$A \wedge (B \wedge C)$' are equivalent in SL. Again, it is easy to test for equivalence using truth tables. Consider the sentences '$\neg(P \vee Q)$' and '$\neg P \wedge \neg Q$'. Are they equivalent? To find out, we construct a truth table.

P	Q	$\neg(P \vee Q)$	$\neg P \wedge \neg Q$
T	T	**F** T T T	F T **F** F T
T	F	**F** T T F	F T **F** T F
F	T	**F** F T T	T F **F** F T
F	F	**T** F F F	T F **T** T F

Look at the columns for the main logical operators; negation for the first sentence, conjunction for the second. On the first three rows, both are false. On the final row, both are true. Since they match on every row, the two sentences are equivalent.

11.3 Satisfiability

In §3, we said that sentences are jointly possible iff it is possible for all of them to be true at once. We can offer a surrogate for this notion too:

> $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are SATISFIABLE (in SL) iff there is some valuation which makes them all true.

Derivatively, sentences are unsatisfiable in SL if there is no valuation that makes them all true. Again, it is easy to test for satisfiability in SL using truth tables.

11.4 Entailment and validity

The following idea is closely related to that of satisfiability in SL:

> The sentences $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ ENTAIL the sentence $\mathscr{C}$ (in SL) iff there is no valuation of the sentence letters which makes all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ true and $\mathscr{C}$ false.

Again, it is easy to test this with a truth table. Let us check whether '$\neg L \rightarrow (J \vee L)$' and '$\neg L$' entail '$J$', we simply need to check whether there is any valuation which makes both '$\neg L \rightarrow (J \vee L)$' and '$\neg L$' true whilst making '$J$' false. So we use a truth table:

J	L	$\neg L \rightarrow (J \vee L)$	$\neg L$	J
T	T	F T **T** T T T	**F** T	**T**
T	F	T F **T** T T F	**T** F	**T**
F	T	F T **T** F T T	**F** T	**F**
F	F	T F **F** F F F	**T** F	**F**

The only row on which both'$\neg L \rightarrow (J \vee L)$' and '$\neg L$' are true is the second row, and that is a row on which 'J' is also true. So '$\neg L \rightarrow (J \vee L)$' and '$\neg L$' entail '$J$'.

We now make an important observation:

If $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ entail $\mathscr{C}$, then $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \therefore \mathscr{C}$ is valid.

Here's why. If $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ entail $\mathscr{C}$, then there is no valuation which makes all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ true whilst making $\mathscr{C}$ false. This means that it is *logically impossible* for $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ all to be true whilst $\mathscr{C}$ is false. But this is just what it takes for an argument, with premises $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ and conclusion $\mathscr{C}$, to be valid!

In short, we have a way to test for the validity of English arguments. First, we symbolize them in SL, as having premises $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$, and conclusion $\mathscr{C}$. Then we test for entailment using truth tables.

11.5 The limits of these tests

We have reached an important milestone: a test for the validity of arguments! However, we should not get carried away just yet. It is important to understand the *limits* of our achievement. We will illustrate these limits with three examples.

First, consider the argument:

1. Daisy has four legs. So Daisy has more than two legs.

To symbolize this argument in SL, we would have to use two different sentence letters—perhaps 'F' and 'T'—for the premise and the conclusion respectively. Now, it is obvious that 'F' does not entail 'T'. The English argument surely seems valid, though!

Second, consider the sentence:

2. Jan is neither bald nor not-bald.

To symbolize this sentence in SL, we would offer something like '$\neg J \wedge \neg\neg J$'. This a contradiction (check this with a truth-table), but sentence 2 does not itself seem like a contradiction; for we might have happily go on to add 'Jan is on the borderline of baldness'!

Third, consider the following sentence:

3. It's not the case that, if God exists, She answers malevolent prayers.

Symbolizing this in SL, we would offer something like '$\neg(G \rightarrow M)$'. Now, '$\neg(G \rightarrow M)$' entails 'G' (again, check this with a truth table). So if we symbolize sentence 3 in SL, it seems to entail that God exists. But that's strange: surely even an atheist can accept sentence 3, without contradicting herself!

One lesson of this is that the symbolization of 3 as '$\neg(G \rightarrow M)$' shows that 3 does not express what we intend. Perhaps we should rephrase it as

3. If God exists, She does not answer malevolent prayers.

and symbolize 3 as '$G \rightarrow \neg M$'. Now, if atheists are right, and there is no God, then 'G' is false and so '$G \rightarrow \neg M$' is true, and the puzzle disappears. However, if 'G' is false, '$G \rightarrow M$', i.e. 'If God exists, She answers malevolent prayers', is *also* true!

In different ways, these four examples highlight some of the limits of working with a language (like SL) that can *only* handle truth-functional operators. Moreover, these limits give rise to some interesting questions in philosophical logic. The case of Jan's baldness (or otherwise) raises the general question of what logic we should use when dealing with *vague* discourse. The case of the atheist raises the question of how to deal with the (so-called) *paradoxes of the material conditional*. Part of the purpose of this course is to equip you with the tools to explore these questions of *philosophical logic*. But we have to walk before we can run; we have to become proficient in using SL, before we can adequately discuss its limits, and consider alternatives.

11.6 The double-turnstile

We are going to use the notion of entailment rather a lot in this course. It will help us, then, to introduce a symbol that abbreviates it. Rather than saying that the SL sentences $\mathscr{A}_1, \mathscr{A}_2, \ldots$ and $\mathscr{A}_n$ together entail $\mathscr{C}$, we will abbreviate this by:

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vDash \mathscr{C}$$

The symbol '$\vDash$' is known as *the double-turnstile*, since it looks like a turnstile with two horizontal beams.

Let me be clear. '$\vDash$' is not a symbol of SL. Rather, it is a symbol of our metalanguage, augmented English (recall the difference between object language and metalanguage from §7). So the metalanguage sentence:

- $P, P \rightarrow Q \vDash Q$

is just an abbreviation for the English sentence:

- The SL sentences 'P' and '$P \rightarrow Q$' entail 'Q'

Note that there is no limit on the number of SL sentences that can be mentioned before the symbol '$\vDash$'. Indeed, we can even consider the limiting case:

$$\vDash \mathscr{C}$$

This says that there is no valuation which makes all the sentences mentioned on the left hand side of '$\vDash$' true whilst making $\mathscr{C}$ false. Since *no* sentences are mentioned on the left hand side of '$\vDash$' in this case, this just means that there is no valuation which makes $\mathscr{C}$ false. Otherwise put, it says that every valuation makes $\mathscr{C}$ true. Otherwise put, it says that $\mathscr{C}$ is a tautology. Equally:

$$\mathscr{A} \vDash$$

says that $\mathscr{A}$ is a contradiction.

11.7 '$\vDash$' versus '$\rightarrow$'

We now want to compare and contrast '$\vDash$' and '$\rightarrow$'.

Observe: $\mathscr{A} \vDash \mathscr{C}$ iff there is no valuation of the sentence letters that makes $\mathscr{A}$ true and $\mathscr{C}$ false.

Observe: $\mathscr{A} \rightarrow \mathscr{C}$ is a tautology iff there is no valuation of the sentence letters that makes $\mathscr{A} \rightarrow \mathscr{C}$ false. Since a conditional is true except when its antecedent is true and its consequent false, $\mathscr{A} \rightarrow \mathscr{C}$ is a tautology iff there is no valuation that makes $\mathscr{A}$ true and $\mathscr{C}$ false.

Combining these two observations, we see that $\mathscr{A} \rightarrow \mathscr{C}$ is a tautology iff $\mathscr{A} \vDash \mathscr{C}$. But there is a really, really important difference between '$\vDash$' and '$\rightarrow$':

'$\rightarrow$' is a sentential operator of SL.
'$\vDash$' is a symbol of augmented English.

Indeed, when '$\rightarrow$' is flanked with two SL sentences, the result is a longer SL sentence. By contrast, when we use '$\vDash$', we form a metalinguistic sentence that *mentions* the surrounding SL sentences.

Practice exercises

A. Revisit your answers to §10**A**. Determine which sentences were tautologies, which were contradictions, and which were neither tautologies nor contradictions.

B. Use truth tables to determine whether these sentences are satisfiable in SL, or unsatisfiable in SL:

1. $A \rightarrow A, \neg A \rightarrow \neg A, A \wedge A, A \vee A$
2. $A \vee B, A \rightarrow C, B \rightarrow C$
3. $B \wedge (C \vee A), A \rightarrow B, \neg(B \vee C)$
4. $A \leftrightarrow (B \vee C), C \rightarrow \neg A, A \rightarrow \neg B$

C. Use truth tables to determine whether each argument is valid or invalid.

1. $A \rightarrow A \therefore A$

2. $A \to (A \land \neg A) \therefore \neg A$
3. $A \lor (B \to A) \therefore \neg A \to \neg B$
4. $A \lor B, B \lor C, \neg A \therefore B \land C$
5. $(B \land A) \to C, (C \land A) \to B \therefore (C \land B) \to A$

D. Determine whether each sentence is a tautology, a contradiction, or a contingent sentence, using a complete truth table.

1. $\neg B \land B$
2. $\neg D \lor D$
3. $(A \land B) \lor (B \land A)$
4. $\neg[A \to (B \to A)]$
5. $A \leftrightarrow [A \to (B \land \neg B)]$
6. $[(A \land B) \leftrightarrow B] \to (A \to B)$

E. Determine whether each the following sentences are logically equivalent using complete truth tables. If the two sentences really are logically equivalent, write "equivalent." Otherwise write, "Not equivalent."

1. A and $\neg A$
2. $A \land \neg A$ and $\neg B \leftrightarrow B$
3. $[(A \lor B) \lor C]$ and $[A \lor (B \lor C)]$
4. $A \lor (B \land C)$ and $(A \lor B) \land (A \lor C)$
5. $[A \land (A \lor B)] \to B$ and $A \to B$

F. Determine whether each the following sentences are logically equivalent using complete truth tables. If the two sentences really are equivalent, write "equivalent." Otherwise write, "not equivalent."

1. $A \to A$ and $A \leftrightarrow A$
2. $\neg(A \to B)$ and $\neg A \to \neg B$
3. $A \lor B$ and $\neg A \to B$
4. $(A \to B) \to C$ and $A \to (B \to C)$
5. $A \leftrightarrow (B \leftrightarrow C)$ and $A \land (B \land C)$

G. Determine whether each collection of sentences is satisfiable or unsatisfiable using a complete truth table.

1. $A \land \neg B, \neg(A \to B), B \to A$
2. $A \lor B, A \to \neg A, B \to \neg B$
3. $\neg(\neg A \lor B), A \to \neg C, A \to (B \to C)$
4. $A \to B, A \land \neg B$
5. $A \to (B \to C), (A \to B) \to C, A \to C$

H. Determine whether each collection of sentences is satisfiable or unsatisfiable, using a complete truth table.

1. $\neg B, A \rightarrow B, A$
2. $\neg(A \vee B), A \leftrightarrow B, B \rightarrow A$
3. $A \vee B, \neg B, \neg B \rightarrow \neg A$
4. $A \leftrightarrow B, \neg B \vee \neg A, A \rightarrow B$
5. $(A \vee B) \vee C, \neg A \vee \neg B, \neg C \vee \neg B$

I. Determine whether each argument is valid or invalid, using a complete truth table.

1. $A \rightarrow B, B \therefore A$
2. $A \leftrightarrow B, B \leftrightarrow C \therefore A \leftrightarrow C$
3. $A \rightarrow B, A \rightarrow C \therefore B \rightarrow C$
4. $A \rightarrow B, B \rightarrow A \therefore A \leftrightarrow B$

J. Determine whether each argument is valid or invalid, using a complete truth table.

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

K. Answer each of the questions below and justify your answer.

1. Suppose that $\mathscr{A}$ and $\mathscr{B}$ are logically equivalent. What can you say about $\mathscr{A} \leftrightarrow \mathscr{B}$?
2. Suppose that $(\mathscr{A} \wedge \mathscr{B}) \rightarrow \mathscr{C}$ is neither a tautology nor a contradiction. What can you say about whether $\mathscr{A}, \mathscr{B} \therefore \mathscr{C}$ is valid?
3. Suppose that $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$ are unsatisfiable. What can you say about $(\mathscr{A} \wedge \mathscr{B} \wedge \mathscr{C})$?
4. Suppose that $\mathscr{A}$ is a contradiction. What can you say about whether $\mathscr{A}, \mathscr{B} \vDash \mathscr{C}$?
5. Suppose that $\mathscr{C}$ is a tautology. What can you say about whether $\mathscr{A}, \mathscr{B} \vDash \mathscr{C}$?
6. Suppose that $\mathscr{A}$ and $\mathscr{B}$ are logically equivalent. What can you say about $(\mathscr{A} \vee \mathscr{B})$?
7. Suppose that $\mathscr{A}$ and $\mathscr{B}$ are *not* logically equivalent. What can you say about $(\mathscr{A} \vee \mathscr{B})$?

L. Consider the following principle:

- Suppose $\mathscr{A}$ and $\mathscr{B}$ are logically equivalent. Suppose an argument contains $\mathscr{A}$ (either as a premise, or as the conclusion). The validity of the argument would be unaffected, if we replaced $\mathscr{A}$ with $\mathscr{B}$.

Is this principle correct? Explain your answer.

12 Truth table shortcuts

With practice, you will quickly become adept at filling out truth tables. In this section, we want to give you some permissible shortcuts to help you along the way.

12.1 Working through truth tables

You will quickly find that you do not need to copy the truth value of each sentence letter, but can simply refer back to them. So you can speed things up by writing:

P	Q	$($	P	$\lor$	Q	$)$	$\leftrightarrow$	$\neg$	P
T	T			T			**F**	F	
T	F			T			**F**	F	
F	T			T			**T**	T	
F	F			F			**F**	T	

You also know for sure that a disjunction is true whenever one of the disjuncts is true. So if you find a true disjunct, there is no need to work out the truth values of the other disjuncts. Thus you might offer:

P	Q	$(\neg$	P	$\lor$	$\neg$	$Q)$	$\lor$	$\neg$	P
T	T	F		F	F		**F**	F	
T	F	F		T	T		**T**	F	
F	T						**T**	T	
F	F						**T**	T	

Equally, you know for sure that a conjunction is false whenever one of the conjuncts is false. So if you find a false conjunct, there is no need to work out the truth value of the other conjunct. Thus you might offer:

P	Q	$\neg$	$(P$	$\land$	$\neg$	$Q)$	$\land$	$\neg$	P
T	T						**F**	F	
T	F						**F**	F	
F	T	T		F			**T**	T	
F	F	T		F			**T**	T	

A similar short cut is available for conditionals. You immediately know that a conditional is true if either its consequent is true, or its antecedent is false. Thus you might present:

P	Q	$((P\to Q)$	$\to P)$	$\to P$
T	T			**T**
T	F			**T**
F	T	T	F	**T**
F	F	T	F	**T**

So '$((P \to Q) \to P) \to P$' is a tautology. In fact, it is an instance of *Peirce's Law*, named after Charles Sanders Peirce.

12.2 Testing for validity and entailment

When we use truth tables to test for validity or entailment, we are checking for *bad* lines: lines where the premises are all true and the conclusion is false. Note:

- Any line where the conclusion is true is not a bad line.
- Any line where some premise is false is not a bad line.

Since *all* we are doing is looking for bad lines, we should bear this in mind. So: if we find a line where the conclusion is true, we do not need to evaluate anything else on that line: that line definitely isn't bad. Likewise, if we find a line where some premise is false, we do not need to evaluate anything else on that line.

With this in mind, consider how we might test the following for validity:

$$\neg L \to (J \lor L), \neg L \therefore J$$

The *first* thing we should do is evaluate the conclusion. If we find that the conclusion is *true* on some line, then that is not a bad line. So we can simply ignore the rest of the line. So at our first stage, we are left with something like:

J	L	$\neg L\to(J\lor L)$	$\neg L$	J
T	T			T
T	F			T
F	T	?	?	F
F	F	?	?	F

where the blanks indicate that we are not going to bother doing any more investigation (since the line is not bad) and the question-marks indicate that we need to keep investigating.

The easiest premise to evaluate is the second, so we next do that:

J	L	$\neg L\to(J\lor L)$	$\neg L$	J
T	T			T
T	F			T
F	T		F	F
F	F	?	T	F

Note that we no longer need to consider the third line on the table: it will not be a bad line, because (at least) one of premises is false on that line. Finally, we complete the truth table:

J	L	$\neg L \to (J \vee L)$	$\neg L$	J
T	T			T
T	F			T
F	T		F	F
F	F	T **F** F	T	F

The truth table has no bad lines, so the argument is valid. (Any valuation on which all the premises are true is a valuation on which the conclusion is true.)

It might be worth illustrating the tactic again. Let us check whether the following argument is valid

$$A \vee B, \neg(A \wedge C), \neg(B \wedge \neg D) \therefore (\neg C \vee D)$$

At the first stage, we determine the truth value of the conclusion. Since this is a disjunction, it is true whenever either disjunct is true, so we can speed things along a bit. We can then ignore every line apart from the few lines where the conclusion is false.

A	B	C	D	$A \vee B$	$\neg(A \wedge C)$	$\neg(B \wedge \neg D)$	$(\neg C \vee D)$
T	T	T	T				**T**
T	T	T	F	?	?	?	F **F**
T	T	F	T				**T**
T	T	F	F				T **T**
T	F	T	T				**T**
T	F	T	F	?	?	?	F **F**
T	F	F	T				**T**
T	F	F	F				T **T**
F	T	T	T				**T**
F	T	T	F	?	?	?	F **F**
F	T	F	T				**T**
F	T	F	F				T **T**
F	F	T	T				**T**
F	F	T	F	?	?	?	F **F**
F	F	F	T				**T**
F	F	F	F				T **T**

We must now evaluate the premises. We use shortcuts where we can:

A	B	C	D	$A \vee B$	$\neg(A \wedge C)$	$\neg(B \wedge \neg D)$	$(\neg C \vee D)$
T	T	T	T				**T**
T	T	T	F	**T**	**F** T		F **F**
T	T	F	T				**T**
T	T	F	F				T **T**
T	F	T	T				**T**
T	F	T	F	**T**	**F** T		F **F**
T	F	F	T				**T**
T	F	F	F				T **T**
F	T	T	T				**T**
F	T	T	F	**T**	**T** F	**F** TT	F **F**
F	T	F	T				**T**
F	T	F	F				T **T**
F	F	T	T				**T**
F	F	T	F	**F**			F **F**
F	F	F	T				**T**
F	F	F	F				T **T**

If we had used no shortcuts, we would have had to write 256 'T's or 'F's on this table. Using shortcuts, we only had to write 37. We have saved ourselves a *lot* of work.

We have been discussing shortcuts in testing for logically validity, but exactly the same shortcuts can be used in testing for entailment. By employing a similar notion of *bad* lines, you can save yourself a huge amount of work.

Practice exercises

A. Using shortcuts, determine whether each sentence is a tautology, a contradiction, or neither.

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

13 Partial truth tables

Sometimes, we do not need to know what happens on every line of a truth table. Sometimes, just a line or two will do.

Tautology. In order to show that a sentence is a tautology, we need to show that it is true on every valuation. That is to say, we need to know that it comes out true on every line of the truth table. 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 false. Therefore, in order to show that some sentence is not a tautology, it is enough to provide a single valuation—a single line of the truth table—which makes the sentence false.

Suppose that we want to show that the sentence '$(U \land T) \to (S \land W)$' is *not* a tautology. We set up a PARTIAL TRUTH TABLE:

S	T	U	W	$(U$	$\land$	$T)$	$\to$	$(S$	$\land$	$W)$
							F			

We have only left space for one line, rather than 16, since we are only looking for one line on which the sentence is false. For just that reason, we have filled in 'F' for the entire sentence.

The main logical operator of the sentence is a conditional. In order for the conditional to be false, the antecedent must be true and the consequent must be false. So we fill these in on the table:

S	T	U	W	$(U$	$\land$	$T)$	$\to$	$(S$	$\land$	$W)$
					T		**F**		F	

In order for the '$(U \land T)$' to be true, both 'U' and 'T' must be true.

S	T	U	W	$(U$	$\land$	$T)$	$\to$	$(S$	$\land$	$W)$
	T	T		T	T	T	**F**		F	

Now we just need to make '$(S \land W)$' 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	$(U \wedge T) \rightarrow (S \wedge W)$
F	T	T	F	T T T **F** F F F

We now have a partial truth table, which shows that '$(U \wedge T) \rightarrow (S \wedge W)$' is not a tautology. Put otherwise, we have shown that there is a valuation which makes '$(U \wedge T) \rightarrow (S \wedge W)$' false, namely, the valuation which makes 'S' false, 'T' true, 'U' true and 'W' false.

Contradiction. Showing that something is a contradiction requires a complete truth table: we need to show that there is no valuation which makes the sentence true; that is, we need to show that the sentence is false on every line of the truth table.

However, to show that something is *not* a contradiction, all we need to do is find a valuation which makes the sentence true, and a single line of a truth table will suffice. We can illustrate this with the same example.

S	T	U	W	$(U \wedge T) \rightarrow (S \wedge W)$
				T

To make the sentence true, it will suffice to ensure that the antecedent is false. Since the antecedent is a conjunction, we can just make one of them false. For no particular reason, we choose to make 'U' false; and then we can assign whatever truth value we like to the other sentence letters.

S	T	U	W	$(U \wedge T) \rightarrow (S \wedge W)$
F	T	F	F	F F T **T** F F F

Truth functional equivalence. To show that two sentences are logically equivalent, we must show that the sentences have the same truth value on every valuation. So this requires a complete truth table.

To show that two sentences are *not* logically equivalent, we only need to show that there is a valuation on which they have different truth values. So this requires only a one-line partial truth table: make the table so that one sentence is true and the other false.

Satisfiability. To show that some sentences are satisfiable, we must show that there is a valuation which makes all of the sentences true,so this requires only a partial truth table with a single line.

To show that some sentences are unsatisfiable, we must show that there is no valuation which makes all of the sentence true. So this requires a complete truth table: You must show that on every row of the table at least one of the sentences is false.

Validity. To show that an argument is valid, we must show that there is no valuation which makes all of the premises true and the conclusion false. So this requires a complete truth table. (Likewise for entailment.)

To show that argument is *invalid*, we must show that there is a valuation which makes all of the premises true and the conclusion false. So this requires only a one-line partial truth table on which all of the premises are true and the conclusion is false. (Likewise for a failure of entailment.)

This table summarises what is required:

	Yes	**No**
tautology?	complete	one-line partial
contradiction?	complete	one-line partial
equivalent?	complete	one-line partial
satisfiable?	one-line partial	complete
valid?	complete	one-line partial
entailment?	complete	one-line partial

Practice exercises

A. Use complete or partial truth tables (as appropriate) to determine whether these pairs of sentences are logically equivalent:

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. $(A \rightarrow B), (\neg B \rightarrow \neg A)$

B. Use complete or partial truth tables (as appropriate) to determine whether these sentences are satisfiable or unsatisfiable:

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

C. Use complete or partial truth tables (as appropriate) to determine whether each argument is valid or invalid:

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

5. $A \leftrightarrow B, B \leftrightarrow C \therefore A \leftrightarrow C$

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

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

F. Determine whether each the following pairs of sentences are logically equivalent using complete truth tables. If the two sentences really are logically equivalent, write "equivalent." Otherwise write, "not equivalent."

1. A and $A \vee A$
2. A and $A \wedge A$
3. $A \vee \neg B$ and $A \rightarrow B$
4. $(A \rightarrow B)$ and $(\neg B \rightarrow \neg A)$
5. $\neg(A \wedge B)$ and $\neg A \vee \neg B$
6. $((U \rightarrow (X \vee X)) \vee U)$ and $\neg(X \wedge (X \wedge U))$
7. $((C \wedge (N \leftrightarrow C)) \leftrightarrow C)$ and $(\neg\neg\neg N \rightarrow C)$
8. $[(A \vee B) \wedge C]$ and $[A \vee (B \wedge C)]$

9. $((L \wedge C) \wedge I)$ and $L \vee C$

G. Determine whether each collection of sentences is satisfiable or unsatisfiable. 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. $A \rightarrow \neg A, \neg A \rightarrow A$
3. $A \vee B, A \rightarrow C, B \rightarrow C$
4. $A \vee B, A \rightarrow C, B \rightarrow C, \neg C$
5. $B \wedge (C \vee A), A \rightarrow B, \neg(B \vee C)$
6. $(A \leftrightarrow B) \rightarrow B, B \rightarrow \neg(A \leftrightarrow B), A \vee B$
7. $A \leftrightarrow (B \vee C), C \rightarrow \neg A, A \rightarrow \neg B$
8. $A \leftrightarrow B, \neg B \vee \neg A, A \rightarrow B$
9. $A \leftrightarrow B, A \rightarrow C, B \rightarrow D, \neg(C \vee D)$
10. $\neg(A \wedge \neg B), B \rightarrow \neg A, \neg B$

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

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

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

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

PART IV

Natural deduction for SL

14 | The very idea of natural deduction

Way back in §2, we said that an argument is valid iff it is impossible to make all of the premises true and the conclusion false.

In the case of SL, this led us to develop truth tables. Each line of a complete truth table corresponds to a valuation. So, when faced with a SL argument, we have a very direct way to assess whether it is possible to make all of the premises true and the conclusion false: just thrash through the truth table.

However, truth tables do not necessarily give us much *insight*. Consider two arguments in SL:

$$P \lor Q, \neg P \therefore Q$$
$$P \to Q, P \therefore Q$$

Clearly, these are valid arguments. You can confirm that they are valid by constructing four-line truth tables, but we might say that they make use of different *forms* of reasoning. It might be nice to keep track of these different forms of inference.

One aim of a *natural deduction system* is to show that particular arguments are valid, in a way that allows us to understand the reasoning that the arguments might involve. We begin with very basic rules of inference. These rules can be combined to offer more complicated arguments. Indeed, with just a small starter pack of rules of inference, we hope to capture all valid arguments.

This is a very different way of thinking about arguments.

With truth tables, we directly consider different ways to make sentences true or false. With natural deduction systems, we manipulate sentences in accordance with rules that we have set down as good rules. The latter promises to give us a better insight—or at least, a different insight—into how arguments work.

The move to natural deduction might be motivated by more than the search for insight. It might also be motivated by *necessity*. Consider:

$$A_1 \to C_1 \therefore (A_1 \land A_2 \land A_3 \land A_4 \land A_5) \to (C_1 \lor C_2 \lor C_3 \lor C_4 \lor C_5)$$

To test this argument for validity, you might use a 1024-line truth table. If you do it correctly, then you will see that there is no line on which all the premises are true and on which the conclusion is false. So you will know that the argument is valid. (But, as just mentioned, there is a sense in which you will not know *why* the argument is valid.) But now consider:

$$A_1 \rightarrow C_1 \therefore (A_1 \wedge A_2 \wedge A_3 \wedge A_4 \wedge A_5 \wedge A_6 \wedge A_7 \wedge A_8 \wedge A_9 \wedge A_{10}) \rightarrow (C_1 \vee C_2 \vee C_3 \vee C_4 \vee C_5 \vee C_6 \vee C_7 \vee C_8 \vee C_9 \vee C_{10})$$

This argument is also valid—as you can probably tell—but to test it requires a truth table with $2^{20} = 1048576$ lines. In principle, we can set a machine to grind through truth tables and report back when it is finished. In practice, complicated arguments in SL can become *intractable* if we use truth tables.

When we get to first-order logic (FOL) (beginning in chapter 21), though, the problem gets dramatically worse. There is nothing like the truth table test for FOL. To assess whether or not an argument is valid, we have to reason about *all* interpretations, but, as we will see, there are infinitely many possible interpretations. We cannot even in principle set a machine to grind through infinitely many possible interpretations and report back when it is finished: it will *never* finish. We either need to come up with some more efficient way of reasoning about all interpretations, or we need to look for something different.

There are, indeed, systems that codify ways to reason about all possible interpretations. They were developed in the 1950s by Evert Beth and Jaakko Hintikka, but we will not follow this path. We will, instead, look to natural deduction.

Rather than reasoning directly about all valuations (in the case of SL), we will try to select a few basic rules of inference. Some of these will govern the behaviour of the sentential operators. Others will govern the behaviour of the quantifiers and identity that are the hallmarks of FOL. The resulting system of rules will give us a new way to think about the validity of arguments. The modern development of natural deduction dates from simultaneous and unrelated papers by Gerhard Gentzen and Stanisław Jaśkowski (1934). However, the natural deduction system that we will consider is based largely around work by Frederic Fitch (first published in 1952).

15 | Basic rules for SL

We will develop a NATURAL DEDUCTION system. For each operator, there will be INTRODUCTION rules, that allow us to prove a sentence that has that operator as the main operator, and ELIMINATION rules, that allow us to prove something given a sentence that has that operator as the main operator.

15.1 The idea of a formal proof

A *formal proof* is a sequence of sentences, some of which are marked as being initial assumptions (or premises). The last line of the formal proof is the conclusion. (Henceforth, we will simply call these 'proofs', but you should be aware that there are *informal proofs* too.)

As an illustration, consider:

$$\neg(A \vee B) \mathrel{\therefore} \neg A \wedge \neg B$$

We will start a proof by writing the premise:

$$\begin{array}{r|l} 1 & \neg(A \vee B) \\ \hline \end{array}$$

Note that we have numbered the premise, since we will want to refer back to it. Indeed, every line on a proof is numbered, so that we can refer back to it.

Note also that we have drawn a line underneath the premise. Everything written above the line is an *assumption*. Everything written below the line will either be something which follows from the assumptions, or it will be some new assumption. We are hoping to conclude that '$\neg A \wedge \neg B$'; so we are hoping ultimately to conclude our proof with

$$\begin{array}{r|l} n & \neg A \wedge \neg B \end{array}$$

for some number n. It doesn't matter what line number we end on, but we would obviously prefer a short proof to a long one.

Similarly, suppose we wanted to consider:

$$A \vee B, \neg(A \wedge C), \neg(B \wedge \neg D) \mathrel{\therefore} \neg C \vee D$$

The argument has three premises, so we start by writing them all down, numbered, and drawing a line under them:

$$\begin{array}{r|l}
1 & A \vee B \\
2 & \neg(A \wedge C) \\
3 & \neg(B \wedge \neg D) \\ \hline
\end{array}$$

and we are hoping to conclude with some line:

$$\begin{array}{r|l}
n & \neg C \vee D
\end{array}$$

All that remains to do is to explain each of the rules that we can use along the way from premises to conclusion. The rules are broken down by our logical operators.

15.2 Conjunction

Suppose we want to show that Ludwig is both reactionary and libertarian. One obvious way to do this would be as follows: first we show that Ludwig is reactionary; then we show that Ludwig is libertarian; then we put these two demonstrations together, to obtain the conjunction.

Our natural deduction system will capture this thought straightforwardly. In the example given, we might adopt the following symbolization key:

R:	Ludwig is reactionary
L:	Ludwig is libertarian

Perhaps we are working through a proof, and we have obtained 'R' on line 8 and 'L' on line 15. Then on any subsequent line we can obtain '$R \wedge L$' thus:

$$\begin{array}{r|ll}
8 & R & \\
15 & L & \\
 & R \wedge L & \wedge\text{I } 8, 15
\end{array}$$

Note that every line of our proof must either be an assumption, or must be justified by some rule. We cite '∧I 8, 15' here to indicate that the line is obtained by the rule of conjunction introduction (∧I) applied to lines 8 and 15. We could equally well obtain:

$$\begin{array}{r|ll}
8 & R & \\
15 & L & \\
 & L \wedge R & \wedge\text{I } 15, 8
\end{array}$$

with the citation reversed, to reflect the order of the conjuncts. More generally, here is our conjunction introduction rule:

$$\begin{array}{r|ll} m & \mathscr{A} & \\ n & \mathscr{B} & \\ & \mathscr{A} \wedge \mathscr{B} & \wedge\text{I } m, n \end{array}$$

To be clear, the statement of the rule is *schematic*. It is not itself a proof. '$\mathscr{A}$' and '$\mathscr{B}$' are not sentences of SL. Rather, they are symbols in the metalanguage, which we use when we want to talk about any sentence of SL (see §7). Similarly, 'm' and 'n' are not a numerals that will appear on any actual proof. Rather, they are symbols in the metalanguage, which we use when we want to talk about any line number of any proof. In an actual proof, the lines are numbered '1', '2', '3', and so forth, but when we define the rule, we use variables to emphasize that the rule may be applied at any point. The rule requires only that we have both conjuncts available to us somewhere in the proof. They can be separated from one another, and they can appear in any order.

The rule is called 'conjunction *introduction*' because it introduces the symbol '$\wedge$' into our proof where it may have been absent. Correspondingly, we have a rule that *eliminates* that symbol. Suppose you have shown that Ludwig is both reactionary and libertarian. You are entitled to conclude that Ludwig is reactionary. Equally, you are entitled to conclude that Ludwig is libertarian. Putting this together, we obtain our conjunction elimination rule(s):

$$\begin{array}{r|ll} m & \mathscr{A} \wedge \mathscr{B} & \\ & \mathscr{A} & \wedge\text{E } m \end{array}$$

and equally:

$$\begin{array}{r|ll} m & \mathscr{A} \wedge \mathscr{B} & \\ & \mathscr{B} & \wedge\text{E } m \end{array}$$

The point is simply that, when you have a conjunction on some line of a proof, you can obtain either of the conjuncts by $\wedge$E. (But one point is worth emphasising: you can only apply this rule when conjunction is the main operator. So you cannot infer 'D' just from '$C \vee (D \wedge E)$'!)

Even with just these two rules, we can start to see some of the power of our formal proof system. Consider:

$$\begin{array}{rl} & [(A \vee B) \to (C \vee D)] \wedge [(E \vee F) \to (G \vee H)] \\ \therefore & [(E \vee F) \to (G \vee H)] \wedge [(A \vee B) \to (C \vee D)] \end{array}$$

The main operator in both the premise and conclusion of this argument is '$\wedge$'. In order to provide a proof, we begin by writing down the premise, which is our assumption. We draw a line below this: everything after this line must follow from our assumptions by (repeated applications of) our rules of inference. So the beginning of the proof looks like this:

$$\begin{array}{r|l}
1 & [(A \vee B) \rightarrow (C \vee D)] \wedge [(E \vee F) \rightarrow (G \vee H)] \\ \hline
\end{array}$$

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

$$\begin{array}{r|ll}
1 & [(A \vee B) \rightarrow (C \vee D)] \wedge [(E \vee F) \rightarrow (G \vee H)] & \\ \hline
2 & [(A \vee B) \rightarrow (C \vee D)] & \wedge\text{E } 1 \\
3 & [(E \vee F) \rightarrow (G \vee H)] & \wedge\text{E } 1
\end{array}$$

So by applying the ∧I rule to lines 3 and 2 (in that order), we arrive at the desired conclusion. The finished proof looks like this:

$$\begin{array}{r|ll}
1 & [(A \vee B) \rightarrow (C \vee D)] \wedge [(E \vee F) \rightarrow (G \vee H)] & \\ \hline
2 & [(A \vee B) \rightarrow (C \vee D)] & \wedge\text{E } 1 \\
3 & [(E \vee F) \rightarrow (G \vee H)] & \wedge\text{E } 1 \\
4 & [(E \vee F) \rightarrow (G \vee H)] \wedge [(A \vee B) \rightarrow (C \vee D)] & \wedge\text{I } 3, 2
\end{array}$$

This is a very simple proof, but it shows how we can chain rules of proof together into longer proofs. In passing, note that investigating this argument with a truth table would have required 256 lines; our formal proof required only four lines.

It is worth giving another example. Back in §10.3, we noted that this argument is valid:

$$A \wedge (B \wedge C) \therefore (A \wedge B) \wedge C$$

To provide a proof corresponding with this argument, we start by writing:

$$\begin{array}{r|l}
1 & A \wedge (B \wedge C) \\ \hline
\end{array}$$

From the premise, we can get each of the conjuncts by applying ∧E twice. We can then apply ∧E twice more, so our proof looks like:

$$\begin{array}{r|ll}
1 & A \wedge (B \wedge C) & \\ \hline
2 & A & \wedge\text{E } 1 \\
3 & B \wedge C & \wedge\text{E } 1 \\
4 & B & \wedge\text{E } 3 \\
5 & C & \wedge\text{E } 3
\end{array}$$

But now we can merrily reintroduce conjunctions in the order we wanted them, so that our final proof is:

$$\begin{array}{r|ll}
1 & A \land (B \land C) & \\ \hline
2 & A & \land\text{E } 1 \\
3 & B \land C & \land\text{E } 1 \\
4 & B & \land\text{E } 3 \\
5 & C & \land\text{E } 3 \\
6 & A \land B & \land\text{I } 2, 4 \\
7 & (A \land B) \land C & \land\text{I } 6, 5
\end{array}$$

Recall that our official definition of sentences in SL only allowed conjunctions with two conjuncts. The proof just given suggests that we could drop inner brackets in all of our proofs. However, this is not standard, and we will not do this. Instead, we will maintain our more austere bracketing conventions. (Though we will still allow ourselves to drop outermost brackets, for legibility.)

Let me offer one final illustration. When using the ∧I rule, there is no need to apply it to different sentences. So, if we want, we can formally prove '*A*' from '*A*' thus:

$$\begin{array}{r|ll}
1 & A & \\ \hline
2 & A \land A & \land\text{I } 1, 1 \\
3 & A & \land\text{E } 2
\end{array}$$

Simple, but effective.

15.3 Conditional

Consider the following argument:

If Jane is smart then she is fast. Jane is smart. ∴ Jane is fast.

This argument is certainly valid, and it suggests a straightforward conditional elimination rule (→E):

$$\begin{array}{r|ll}
m & \mathscr{A} \to \mathscr{B} & \\
n & \mathscr{A} & \\
 & \mathscr{B} & \to\text{E } m, n
\end{array}$$

This rule is also sometimes called *modus ponens*. Again, this is an elimination rule, because it allows us to obtain a sentence that may not contain '→', having started with a sentence that did contain '→'. Note that the conditional and the antecedent can be separated from one another, and they can appear in any order. However, in the citation for →E, we always cite the conditional first, followed by the antecedent.

The rule for conditional introduction is also quite easy to motivate. The following argument should be valid:

> Ludwig is reactionary. Therefore if Ludwig is libertarian, then Ludwig is both reactionary *and* libertarian.

If someone doubted that this was valid, we might try to convince them otherwise by explaining ourselves as follows:

> Assume that Ludwig is reactionary. Now, *additionally* assume that Ludwig is libertarian. Then by conjunction introduction—which we just discussed—Ludwig is both reactionary and libertarian. Of course, that's conditional on the assumption that Ludwig is libertarian. But this just means that, if Ludwig is libertarian, then he is both reactionary and libertarian.

Transferred into natural deduction format, here is the pattern of reasoning that we just used. We started with one premise, 'Ludwig is reactionary', thus:

```
1 | R
  |___
```

The next thing we did is to make an *additional* assumption ('Ludwig is libertarian'), for the sake of argument. To indicate that we are no longer dealing *merely* with our original assumption ('*R*'), but with some additional assumption, we continue our proof as follows:

```
1 | R
  |___
2 |  | L
  |  |___
```

Note that we are *not* claiming, on line 2, to have proved '*L*' from line 1, so we do not need to write in any justification for the additional assumption on line 2. We do, however, need to mark that it is an additional assumption. We do this by drawing a line under it (to indicate that it is an assumption) and by indenting it with a further vertical line (to indicate that it is additional).

With this extra assumption in place, we are in a position to use ∧I. So we can continue our proof:

```
1 | R
  |___
2 |  | L
  |  |___
3 |  | R ∧ L      ∧I 1, 2
```

So we have now shown that, on the additional assumption, 'L', we can obtain '$R \wedge L$'. We can therefore conclude that, if 'L' obtains, then so does '$R \wedge L$'. Or, to put it more briefly, we can conclude '$L \to (R \wedge L)$':

$$\begin{array}{r|l|ll}
1 & R & & \\ \cline{2-2}
2 & & L & \\ \cline{3-3}
3 & & R \wedge L & \wedge\text{I } 1, 2 \\
4 & L \to (R \wedge L) & & \to\text{I } 2\text{–}3
\end{array}$$

Observe that we have dropped back to using one vertical line. We have *discharged* the additional assumption, 'L', since the conditional itself follows just from our original assumption, 'R'.

The general pattern at work here is the following. We first make an additional assumption, A; and from that additional assumption, we prove B. In that case, we know the following: If A, then B. This is wrapped up in the rule for conditional introduction:

$$\begin{array}{r|l|ll}
i & & \mathscr{A} & \\ \cline{3-3}
j & & \mathscr{B} & \\
 & \mathscr{A} \to \mathscr{B} & & \to\text{I } i\text{–}j
\end{array}$$

There can be as many or as few lines as you like between lines i and j.

It will help to offer a second illustration of →I in action. Suppose we want to consider the following:

$$P \to Q, Q \to R \therefore P \to R$$

We start by listing *both* of our premises. Then, since we want to arrive at a conditional (namely, '$P \to R$'), we additionally assume the antecedent to that conditional. Thus our main proof starts:

$$\begin{array}{r|l|l}
1 & P \to Q & \\
2 & Q \to R & \\ \cline{2-3}
3 & & P \\ \cline{3-3}
\end{array}$$

Note that we have made 'P' available, by treating it as an additional assumption, but now, we can use →E on the first premise. This will yield 'Q'. We can then use →E on the second premise. So, by assuming 'P' we were able to prove 'R', so we apply the →I rule—discharging 'P'—and finish the proof. Putting all this together, we have:

```
1 │ P → Q
2 │ Q → R
  ├──────
3 │ │ P
  │ ├───
4 │ │ Q        →E 1, 3
5 │ │ R        →E 2, 4
6 │ P → R      →I 3–5
```

15.4 Additional assumptions and subproofs

The rule →I invoked the idea of making additional assumptions. These need to be handled with some care.

Consider this proof:

```
1 │ A
  ├───
2 │ │ B
  │ ├───
3 │ │ B ∧ B     ∧I 2, 2
4 │ │ B         ∧E 3
5 │ B → B       →I 2–4
```

This is perfectly in keeping with the rules we have laid down already, and it should not seem particularly strange. Since '$B \rightarrow B$' is a tautology, no particular premises should be required to prove it.

But suppose we now tried to continue the proof as follows:

```
1 │ A
  ├───
2 │ │ B
  │ ├───
3 │ │ B ∧ B     ∧I 2, 2
4 │ │ B         ∧E 3
5 │ B → B       →I 2–4
6 │ B           naughty attempt to invoke →E 5, 4
```

If we were allowed to do this, it would be a disaster. It would allow us to prove any sentence letter from any other sentence letter. However, if you tell me that Anne is fast (symbolized by 'A'), we shouldn't be able to conclude that Queen Boudica stood twenty-feet tall (symbolized by 'B')! We must be prohibited from doing this, but how are we to implement the prohibition?

We can describe the process of making an additional assumption as one of performing a *subproof*: a subsidiary 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 the assumption upon which the subproof will be based. A subproof can be thought of as essentially posing this question: *what could we show, if we also make this additional assumption?*

When we are working within the subproof, we can refer to the additional assumption that we made in introducing the subproof, and to anything that we obtained from our original assumptions. (After all, those original assumptions are still in effect.) At some point though, we will want to stop working with the additional assumption: we will want to return from the subproof to the main proof. To indicate that we have returned to the main proof, the vertical line for the subproof comes to an end. At this point, we say that the subproof is CLOSED. Having closed a subproof, we have set aside the additional assumption, so it will be illegitimate to draw upon anything that depends upon that additional assumption. Thus we stipulate:

To cite individual lines when applying a rule, those lines must (1) come before the application of the rule, but (2) not occur within a closed subproof.

This stipulation rules out the disastrous attempted proof above. The rule of $\rightarrow$E requires that we cite two individual lines from earlier in the proof. In the purported proof, above, one of these lines (namely, line 4) occurs within a subproof that has (by line 6) been closed. This is illegitimate.

Closing a subproof is called DISCHARGING the assumptions of that subproof. So we can put the point this way: *you cannot refer back to anything that was obtained using discharged assumptions*.

Subproofs, then, allow us to think about what we could show, if we made additional assumptions. The point to take away from this is not surprising—in the course of a proof, we have to keep very careful track of what assumptions we are making, at any given moment. Our proof system does this very graphically. (Indeed, that's precisely why we have chosen to use *this* proof system.)

Once we have started thinking about what we can show by making additional assumptions, nothing stops us from posing the question of what we could show if we were to make *even more* assumptions. This might motivate us to introduce a subproof within a subproof. Here is an example which only uses the rules of proof that we have considered so far:

```
1 | A
  |---
2 | | B
  | |---
3 | | | C
  | | |---
4 | | | A ∧ B                ∧I 1, 2
5 | | C → (A ∧ B)            →I 3–4
6 | B → (C → (A ∧ B))        →I 2–5
```

Notice that the citation on line 4 refers back to the initial assumption (on line 1) and an assumption of a subproof (on line 2). This is perfectly in order, since neither assumption has been discharged at the time (i.e. by line 4).

Again, though, we need to keep careful track of what we are assuming at any given moment. Suppose we tried to continue the proof as follows:

```
1 | A
  |---
2 | | B
  | |---
3 | | | C
  | | |---
4 | | | A ∧ B                ∧I 1, 2
5 | | C → (A ∧ B)            →I 3–4
6 | B → (C → (A ∧ B))        →I 2–5
7 | C → (A ∧ B)              naughty attempt to invoke →I 3–4
```

This would be awful. If we tell you that Anne is smart, you should not be able to infer that, if Cath is smart (symbolized by 'C') then *both* Anne is smart and Queen Boudica stood 20-feet tall! But this is just what such a proof would suggest, if it were permissible.

The essential problem is that the subproof that began with the assumption 'C' depended crucially on the fact that we had assumed 'B' on line 2. By line 6, we have *discharged* the assumption 'B': we have stopped asking ourselves what we could show, if we also assumed 'B'. So it is simply cheating, to try to help ourselves (on line 7) to the subproof that began with the assumption 'C'. Thus we stipulate, much as before:

> To cite a subproof when applying a rule, the subproof must (1) come before the application of the rule, but (2) not occur within some other closed subproof.

The attempted disastrous proof violates this stipulation. The subproof of lines 3–4 occurs within a subproof that ends on line 5. So it cannot be invoked in line 7. The attempted disastrous proof violates this stipulation. The subproof of lines 3–4 occurs within the subproof of lines 2–5, so the subproof of lines 3–4 cannot be invoked in line 7.

It is always permissible to open a subproof with any assumption. However, 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 obtain a conditional by →I, for instance, you must assume the antecedent of the conditional in a subproof.

Equally, it is always permissible to close a subproof and discharge its assumptions. However, it will not be helpful to do so until you have reached something useful.

15.5 Biconditional

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

In order to prove '$W \leftrightarrow X$', for instance, you must be able to prove 'X' on the assumption 'W' *and* prove 'W' on the assumption 'X'. The biconditional introduction rule (↔I) therefore requires two subproofs. Schematically, the rule works like this:

$$
\begin{array}{l|l|ll}
i & & \underline{\mathscr{A}} & \\
j & & \mathscr{B} & \\
k & & \underline{\mathscr{B}} & \\
l & & \mathscr{A} & \\
 & \multicolumn{2}{l}{\mathscr{A} \leftrightarrow \mathscr{B}} & \leftrightarrow\text{I } i\text{–}j,\ k\text{–}l
\end{array}
$$

There can be as many lines as you like between i and j, and as many lines as you like between k and l. Moreover, the subproofs can come in any order, and the second subproof does not need to come immediately after the first.

The biconditional elimination rule (↔E) lets you do a bit more than the conditional rule. If you have the left-hand subsentence of the biconditional, you can obtain the right-hand subsentence. If you have the right-hand subsentence, you can obtain the left-hand subsentence. So we allow:

$$
\begin{array}{l|ll}
m & \mathscr{A} \leftrightarrow \mathscr{B} & \\
n & \mathscr{A} & \\
 & \mathscr{B} & \leftrightarrow\text{E } m, n
\end{array}
$$

and equally:

$$\begin{array}{r|ll} m & \mathscr{A} \leftrightarrow \mathscr{B} & \\ n & \mathscr{B} & \\ & \mathscr{A} & \leftrightarrow\text{E } m, n \end{array}$$

Note that the biconditional, and the right or left half, can be separated from one another, and they can appear in any order. However, in the citation for ↔E, we always cite the biconditional first.

15.6 Disjunction

Suppose Ludwig is reactionary. Then Ludwig is either reactionary or libertarian. After all, to say that Ludwig is either reactionary or libertarian is to say something weaker than to say that Ludwig is reactionary.

Let me emphasize this point. Suppose Ludwig is reactionary. It follows that Ludwig is *either* reactionary *or* a kumquat. Equally, it follows that *either* Ludwig is reactionary *or* that kumquats are the only fruit. Equally, it follows that *either* Ludwig is reactionary *or* that God is dead. Many of these are strange inferences to draw, but there is nothing *logically* wrong with them (even if they maybe violate all sorts of implicit conversational norms).

Armed with all this, we present the disjunction introduction rule(s):

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

and

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

Notice that $\mathscr{B}$ can be *any* sentence whatsoever, so the following is a perfectly acceptable proof:

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

Using a truth table to show this would have taken 128 lines.

The disjunction elimination rule is, though, slightly trickier. Suppose that either Ludwig is reactionary or he is libertarian. What can you conclude? Not that Ludwig is reactionary; it might be that he is libertarian instead. Equally, not that Ludwig is libertarian; for he might merely be reactionary. Disjunctions, just by themselves, are hard to work with.

But suppose that we could somehow show both of the following: first, that Ludwig's being reactionary entails that he is an Austrian economist: second, that Ludwig's being libertarian entails that he is an Austrian economist. Then if we know that Ludwig is either reactionary or libertarian, then we know that, whichever he is, Ludwig is an Austrian economist. This insight can be expressed in the following rule, which is our disjunction elimination (∨E) rule:

$$
\begin{array}{r|lll}
m & \mathscr{A} \lor \mathscr{B} & & \\
i & & \vert\ \underline{\mathscr{A}} & \\
j & & \vert\ \mathscr{C} & \\
k & & \vert\ \underline{\mathscr{B}} & \\
l & & \vert\ \mathscr{C} & \\
 & \mathscr{C} & & \lor\text{E } m, i\text{–}j, k\text{–}l
\end{array}
$$

This is obviously a bit clunkier to write down than our previous rules, but the point is fairly simple. Suppose we have some disjunction, $\mathscr{A} \lor \mathscr{B}$. Suppose we have two subproofs, showing us that $\mathscr{C}$ follows from the assumption that $\mathscr{A}$, and that $\mathscr{C}$ follows from the assumption that $\mathscr{B}$. Then we can infer $\mathscr{C}$ itself. As usual, there can be as many lines as you like between i and j, and as many lines as you like between k and l. Moreover, the subproofs and the disjunction can come in any order, and do not have to be adjacent.

Some examples might help illustrate this. Consider this argument:

$$(P \land Q) \lor (P \land R) \therefore P$$

An example proof might run thus:

$$
\begin{array}{r|lll}
1 & \underline{(P \land Q) \lor (P \land R)} & & \\
2 & & \vert\ \underline{P \land Q} & \\
3 & & \vert\ P & \land\text{E } 2 \\
4 & & \vert\ \underline{P \land R} & \\
5 & & \vert\ P & \land\text{E } 4 \\
6 & P & & \lor\text{E } 1, 2\text{–}3, 4\text{–}5
\end{array}
$$

Here is a slightly harder example. Consider:

$$A \wedge (B \vee C) \therefore (A \wedge B) \vee (A \wedge C)$$

Here is a proof corresponding to this argument:

1	$A \wedge (B \vee C)$	
2	A	$\wedge$E 1
3	$B \vee C$	$\wedge$E 1
4	$\mid B$	
5	$\mid A \wedge B$	$\wedge$I 2, 4
6	$\mid (A \wedge B) \vee (A \wedge C)$	$\vee$I 5
7	$\mid C$	
8	$\mid A \wedge C$	$\wedge$I 2, 7
9	$\mid (A \wedge B) \vee (A \wedge C)$	$\vee$I 8
10	$(A \wedge B) \vee (A \wedge C)$	$\vee$E 3, 4–6, 7–9

Don't be alarmed if you think that you wouldn't have been able to come up with this proof yourself. The ability to come up with novel proofs comes with practice. The key question at this stage is whether, looking at the proof, you can see that it conforms with the rules that we have laid down. That just involves checking every line, and making sure that it is justified in accordance with the rules we have laid down.

15.7 Contradiction and negation

We have only one operator left to deal with: negation. But to tackle it, we must connect negation with *contradiction*.

An effective form of argument is to argue your opponent into contradicting themselves. At that point, you have them on the ropes. They have to give up at least one of their assumptions. We are going to make use of this idea in our proof system, by adding a new symbol, '$\bot$', to our proofs. This should be read as something like 'contradiction!' or 'reductio!' or 'but that's absurd!' The rule for introducing this symbol is that we can use it whenever we explicitly contradict ourselves, i.e. whenever we find both a sentence and its negation appearing in our proof:

$$\begin{array}{r|ll}
m & \neg\mathscr{A} & \\
n & \mathscr{A} & \\
 & \bot & \bot\text{I } m, n
\end{array}$$

It does not matter what order the sentence and its negation appear in, and they do not need to appear on adjacent lines. However, we always cite the line number of the negation first, followed by that of the sentence it is a negation of.

There is obviously a tight link between contradiction and negation. The rule ⊥I lets us proceed from two contradictory sentences—$\mathscr{A}$ and its negation $\neg\mathscr{A}$—to an explicit contradition ⊥.

We have said that '⊥' should be read as something like 'contradiction!' but this does not tell us much about the symbol. There are, roughly, three ways to approach the symbol.

- We might regard '⊥' as a new atomic sentence of SL, but one which can only ever have the truth value False.
- We might regard '⊥' as an abbreviation for some canonical contradiction, such as '$A \wedge \neg A$'. This will have the same effect as the above—obviously, '$A \wedge \neg A$' only ever has the truth value False—but it means that, officially, we do not need to add a new symbol to SL.
- We might regard '⊥', not as a symbol of SL, but as something more like a *punctuation mark* that appears in our proofs. (It is on a par with the line numbers and the vertical lines, say.)

There is something very philosophically attractive about the third option, but here we will *officially* adopt the first. '⊥' is to be read as a sentence letter that is always false. This means that we can manipulate it, in our proofs, just like any other sentence.

We still have to state a rule for negation introduction. The rule is very simple: if assuming something leads you to a contradiction, then the assumption must be wrong. This thought motivates the following rule:

$$\begin{array}{r|l|ll}
i & & \mathscr{A} & \\ \cline{3-3}
j & & \bot & \\
 & \neg\mathscr{A} & & \neg\text{I } i\text{–}j
\end{array}$$

There can be as many lines between i and j as you like. To see this in practice, and interacting with negation, consider this proof:

$$\begin{array}{r|l l}
1 & D & \\
\cline{2-2}
2 & \quad\vert\ \neg D & \\
3 & \quad\vert\ \bot & \bot\text{I } 2, 1 \\
4 & \neg\neg D & \neg\text{I } 2\text{–}3
\end{array}$$

If the assumption that $\mathscr{A}$ is true leads to a contradiction, $\mathscr{A}$ cannot be true, i.e. it must be false, i.e., $\neg\mathscr{A}$ must be true. Of course, if the assumption that $\mathscr{A}$ is false (i.e. the assumption that $\neg\mathscr{A}$ is true) leads to a contradiction, then $\mathscr{A}$ cannot be false, i.e. $\mathscr{A}$ must be true. So we can consider the following rule:

$$\begin{array}{r|l l}
i & \quad\vert\ \neg\mathscr{A} & \\
j & \quad\vert\ \bot & \\
 & \mathscr{A} & \neg\text{E } i\text{–}j
\end{array}$$

This rule is called *negation elimination*, since it allows us to eliminate the negation from $\neg\mathscr{A}$, by showing that assuming $\neg\mathscr{A}$ leads to a contradiction. Formally, the rule is very similar to $\neg$I, but $\mathscr{A}$ and $\neg\mathscr{A}$ have changed places.[1]

Using $\neg$I, we were able to give a proof of $\neg\neg D$ from D. Using $\neg$E, we can go the other direction (with essentially the same proof).

$$\begin{array}{r|l l}
1 & \neg\neg D & \\
\cline{2-2}
2 & \quad\vert\ \neg D & \\
3 & \quad\vert\ \bot & \bot\text{I } 1, 2 \\
4 & D & \neg\text{E } 2\text{–}3
\end{array}$$

We need one last rule. It is an elimination rule for '$\bot$', sometimes called *explosion*.[2] If we obtain a contradiction, symbolized by '$\bot$', then we can infer whatever we like. How can this be motivated, as a rule of argumentation? Well, consider the English rhetorical device '…and if *that's* true, I'll eat my hat'. Since contradictions simply cannot be true, if one *is* true then not only will I eat my hat, I'll have it too.[3] Here is the formal rule:

[1] There are logicians who have qualms about $\neg$E, but not about $\neg$I. They are called "intuitionists." Intuitionists don't buy our basic assumption that every sentence has one of two truth values, true or false. They also think that $\neg$ works differently—for them, a proof of $\bot$ from $\mathscr{A}$ guarantees $\neg\mathscr{A}$, but a proof of $\bot$ from $\neg\mathscr{A}$ does not guarantee that $\mathscr{A}$, but only $\neg\neg\mathscr{A}$. So, for them, $\mathscr{A}$ and $\neg\neg\mathscr{A}$ are not equivalent.

[2] The latin name for this principle is*ex contradictione quod libet*, "from contradiction, anything."

[3] Thanks to Adam Caulton for this.

$$\begin{array}{r|ll} m & \bot & \\ & \mathscr{A} & \bot\text{E } m \end{array}$$

Note that $\mathscr{A}$ can be *any* sentence whatsoever.

The $\bot$-elimination rule is a bit odd. It looks like $\mathscr{A}$ arrives in our proof like a bunny out of a hat. When trying to find proofs, it is very tempting to try to use it everywhere, since it seems so powerful. Resist this temptation: you can only apply it when you already have $\bot$! And you get $\bot$ only when your assumptions are contradictory.

Still, isn't it odd that from a contradiction anything whatsoever should follow? Not according to our notion of entailment and validity. For $\mathscr{A}$ entails $\mathscr{B}$ iff there is no valuation of the sentence letters which makes $\mathscr{A}$ true and $\mathscr{B}$ false at the same time. Now $\bot$ is a contradiction—it is never true, whatever the valuation of the sentence letters. Since there is no valuation which makes $\bot$ true, there of course is also no valuation that makes $\bot$ true and $\mathscr{B}$ false! So according to our definition of entailment, $\bot \vDash \mathscr{B}$, whatever $\mathscr{B}$ is. A contradiction entails anything.[4]

These are all of the basic rules for the proof system for SL.

Practice exercises

A. The following two 'proofs' are *incorrect*. Explain the mistakes they make.

$$\begin{array}{r|l|ll}
1 & \multicolumn{2}{l}{(\neg L \wedge A) \vee L} & \\ \cline{2-3}
2 & & \neg L \wedge A & \\ \cline{3-3}
3 & & \neg L & \wedge\text{E } 3 \\
4 & & A & \wedge\text{E } 1 \\
5 & & L & \\ \cline{3-3}
6 & & \bot & \bot\text{I } 3, 5 \\
7 & & A & \bot\text{E } 6 \\
8 & \multicolumn{2}{l}{A} & \vee\text{E } 1, 2\text{–}4, 5\text{–}7
\end{array}$$

[4] There are some logicians who don't buy this. They think that if $\mathscr{A}$ entails $\mathscr{B}$, there must be some *relevant connection* between $\mathscr{A}$ and $\mathscr{B}$—and there isn't one between $\bot$ and some arbitrary sentence $\mathscr{B}$. So these logicians develop other, "relevant" logics in which you aren't allowed the explosion rule.

$$\begin{array}{r|ll}
1 & A \wedge (B \wedge C) & \\
2 & (B \vee C) \to D & \\
\hline
3 & B & \wedge\text{E } 1 \\
4 & B \vee C & \vee\text{I } 3 \\
5 & D & \to\text{E } 4, 2
\end{array}$$

B. The following three proofs are missing their citations (rule and line numbers). Add them, to turn them into *bona fide* proofs. Additionally, write down the argument that corresponds to each proof.

$$\begin{array}{r|l}
1 & P \wedge S \\
2 & S \to R \\
\hline
3 & P \\
4 & S \\
5 & R \\
6 & R \vee E
\end{array}$$

$$\begin{array}{r|l}
1 & \neg L \to (J \vee L) \\
2 & \neg L \\
\hline
3 & J \vee L \\
4 & \quad \big| \; \underline{J} \\
5 & \quad \big| \; J \wedge J \\
6 & \quad \big| \; J \\
7 & \quad \big| \; \underline{L} \\
8 & \quad \big| \; \bot \\
9 & \quad \big| \; J \\
10 & J
\end{array}$$

$$\begin{array}{r|l}
1 & A \to D \\
\hline
2 & \quad \big| \; \underline{A \wedge B} \\
3 & \quad \big| \; A \\
4 & \quad \big| \; D \\
5 & \quad \big| \; D \vee E \\
6 & (A \wedge B) \to (D \vee E)
\end{array}$$

C. Give a proof for each of the following arguments:

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

11. $\neg(P \to Q) \therefore \neg Q$
12. $\neg(P \to Q) \therefore P$

16 | Constructing proofs

There is no simple recipe for finding proofs, and there is no substitute for practice. Here, though, are some rules of thumb and strategies to keep in mind.

16.1 Working backwards from what we want

The ultimate goal is to obtain the conclusion. Look at the conclusion and ask what the introduction rule is for its main 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 get to this new goal.

For example: If your conclusion is a conditional $\mathscr{A} \rightarrow \mathscr{B}$, plan to use the $\rightarrow$I rule. This requires starting a subproof in which you assume $\mathscr{A}$. The subproof ought to end with $\mathscr{B}$. Then, continue by thinking about what can you do to get $\mathscr{B}$ inside that subproof, and how you can use the assumption $\mathscr{A}$.

If your goal is a conjunction, conditional, or negated sentence, you should start by working backwards in this way. We'll describe what you have to do in each of these cases in detail.

If we want to prove $\mathscr{A} \wedge \mathscr{B}$, working backward means we should write $\mathscr{A} \wedge \mathscr{B}$ at the bottom of our proof, and try to prove it using $\wedge$I. At the top, we'll write out the premises of the proof, if there are any. Then, at the bottom, we write the sentence we want to prove. If it is a conjunction, we'll prove it using $\wedge$I.

$$\begin{array}{l|ll}
1 & \mathscr{P}_1 & \\
 & \vdots & \\
k & \underline{\mathscr{P}_k} & \\
 & \vdots & \\
n & \mathscr{A} & \\
 & \vdots & \\
m & \mathscr{B} & \\
m+1 & \mathscr{A} \wedge \mathscr{B} & \wedge\text{I } n, m
\end{array}$$

For $\wedge$I, we need to prove $\mathscr{A}$ first, then prove $\mathscr{B}$. For the last line, we'll cite the lines where we proved $\mathscr{A}$ and $\mathscr{B}$, and use $\wedge$I. The parts of the proof labelled '$\cdots$' have to still be filled in. We'll mark the line numbers 'm, n' for now. When the proof is complete, these placeholders can be replaced by actual numbers.

If our goal is to prove a conditional $\mathscr{A} \rightarrow \mathscr{B}$, we'll have to use $\rightarrow$I. This requires a subproof starting with $\mathscr{A}$ and ending with $\mathscr{B}$. We'll set up our proof as follows:

```
n      | | 𝒜
       | |___
       | | ⋮
m      | | ℬ
m + 1  | 𝒜 → ℬ      →I n–m
```

Again we'll leave placeholders in the line number slots. We'll record the last inference as $\rightarrow$I, citing the subproof.

If we want to prove $\neg\mathscr{A}$, we'll have to use $\neg$I.

```
n      | | 𝒜
       | |___
       | | ⋮
m      | | ⊥
m + 1  | ¬𝒜      ¬I n–m
```

For $\neg$I, we have to start a subproof with assumption $\mathscr{A}$; the last line of the subproof has to be $\bot$. We'll cite the subproof, and use $\neg$I.

When working backwards, continue to do so until as long as you can. So if you're working backwards to prove $\mathscr{A} \rightarrow \mathscr{B}$ and have set up a subproof in which you want to prove $\mathscr{B}$, now look at it $\mathscr{B}$. If, say, it is a conjunction, work backwards from it, and write down the two conjuncts inside your subproof.

Of course, you can also work backward from a disjunction $\mathscr{A} \vee \mathscr{B}$, if that is your goal. The $\vee$I rule requires that you have one of the disjuncts in order to infer $\mathscr{A} \vee \mathscr{B}$. So to work backwards, you pick a disjunct, infer $\mathscr{A} \vee \mathscr{B}$ from it, and then continue to look for a proof of the disjunct you picked:

```
       | ⋮
n      | 𝒜
n + 1  | 𝒜 ∨ ℬ      ∨I n
```

However, you may not be able to prove the disjunct you picked. In that case you have to backtrack. When you can't fill in the $\cdots$, delete everything, and try with the other disjunct:

$$\begin{array}{l|ll}
 & \vdots & \\
n & \mathscr{B} & \\
n+1 & \mathscr{A} \lor \mathscr{B} & \lor\text{I } n
\end{array}$$

Obviously, deleting everything and starting over is frustrating, so you should avoid it. If your goal is a disjunction, therefore, you should *not* start by working backwards: try working forwards first, and apply the ∨I strategy only when working forwards (and working backwards using ∧I, →I, and ¬I) no longer work.

16.2 Work forwards from what you have

Your proof may have premises. If you've worked backwards in order to prove a conditional or a negated sentence, you will have set up subproofs with an assumption. These premises and assumptions are sentences you have and can use to fill in the missing steps in your proof. Using them means applying emlimination rules for the main operators of these sentences. These will tell you what your options are.

If we want to use a sentence of the form $\mathscr{A} \land \mathscr{B}$ we use ∧E. That rule allows us to do two things: infer $\mathscr{A}$, and infer $\mathscr{B}$. So in a proof where we have $\mathscr{A} \land \mathscr{B}$, we can always work forwards by writing $\mathscr{A}$ and $\mathscr{B}$ immediately below the conjunction:

$$\begin{array}{l|ll}
 & \vdots & \\
n & \mathscr{A} \land \mathscr{B} & \\
n+1 & \mathscr{A} & \land\text{E } n \\
n+2 & \mathscr{B} & \land\text{E } n \\
 & \vdots &
\end{array}$$

We'll probably need them further down in the proof. In some cases we won't need both; but it doesn't hurt to write them both down.

Working forwards from a disjunction works a bit differently. To use a disjunction, we use the ∨E rule. But in order to apply that rule, it is not enough to know what the disjuncts of the disjunction are that we want to use. We must also keep in mind what it is that we want to prove. Suppose we want to prove $\mathscr{C}$, and we have $\mathscr{A} \lor B$ to work with. (Again, that $\mathscr{A} \lor B$ may be a premise of the proof, an assumption of a subproof, or something already proved, e.g., by ∧E.) In order to be able to apply the ∨E rule, we'll have to set up two subproofs.

```
        │ ⋮
n       │ 𝒜 ∨ ℬ
n + 1   │ │ 𝒜
        │ ├──
        │ │ ⋮
m       │ │ 𝒞
m + 1   │ │ ℬ
        │ ├──
        │ │ ⋮
k       │ │ 𝒞
k + 1   │ 𝒞          ∨E n, (n + 1)–m, (m + 1)–k
        │ ⋮
```

The first subproof starts with the first disjunct, $\mathscr{A}$, and ends with the sentence we're looking for, $\mathscr{C}$. The second subproof starts with the other disjunct, $\mathscr{B}$, and also ends with the goal sentence $\mathscr{C}$. Each of these subproofs have to be filled in further. We can justify $\mathscr{C}$ by using ∨E, citing the line with $\mathscr{A} \lor \mathscr{B}$ and the two subproofs.

In order to use a conditional $\mathscr{A} \rightarrow \mathscr{B}$, you need the antecedent $\mathscr{A}$ in order to apply →E. So to work forward from a conditional, you will derive $\mathscr{B}$, justify it by →E, and set up $\mathscr{A}$ as a new subgoal.

```
        │ ⋮
n       │ 𝒜 → ℬ
        │ ⋮
m       │ 𝒜
m + 1   │ ℬ          →E n, m
        │ ⋮
```

Finally, to use a negated sentence $\neg\mathscr{A}$, you would apply *bot*I. It requires, in addition to $\neg\mathscr{A}$ also the corresponding sentence $\mathscr{A}$ without the negation. The sentence you'll get is always the same: $\bot$. So working forward from a negated sentence works especially well inside a subproof that you'll want to use for ¬I (or ¬E).

$$
\begin{array}{r|ll}
 & \vdots & \\
n & \neg\mathscr{A} & \\
 & \vdots & \\
m & \mathscr{A} & \\
m+1 & \bot & \bot\text{I } n, m \\
 & \vdots &
\end{array}
$$

16.3 Strategies at work

Suppose we want to show that the argument $(A \land B) \lor (A \land C) \therefore A \land (B \lor C)$ is valid. We start the proof by writing the premise and conclusion down. (On a piece of paper, you would want as much space as possible between them, so write the premises at the top of the sheet and the conclusion at the bottom.)

$$
\begin{array}{r|l}
1 & (A \land B) \lor (A \land C) \\
\cline{2-2}
 & \vdots \\
n & A \land (B \lor C)
\end{array}
$$

We now have two options: either work backwards from the conclusion, or work forwards from the premise. We'll pick the second strategy: we use the disjunction on line 1, and set up the subproofs we need for $\lor$E.

$$
\begin{array}{r|l|ll}
1 & \multicolumn{2}{l}{(A \land B) \lor (A \land C)} & \\
\cline{2-3}
2 & & A \land B & \\
\cline{3-3}
 & & \vdots & \\
n & & A \land (B \lor C) & \\
n+1 & & A \land C & \\
\cline{3-3}
 & & \vdots & \\
m & & A \land (B \lor C) & \\
m+1 & \multicolumn{2}{l}{A \land (B \lor C)} & \lor\text{E } 1, 2\text{–}n, n+1\text{–}m
\end{array}
$$

In the first subproof, we now work backwards from the conclusion and set up the sub goals for proving line 6 using $\land$I.

$$\begin{array}{lll}
1 & (A \wedge B) \vee (A \wedge C) & \\
\hline
2 & \quad \mid\ A \wedge B & \\
 & \quad \mid\ \vdots & \\
i & \quad \mid\ A & \\
 & \quad \mid\ \vdots & \\
n-1 & \quad \mid\ B \vee C & \\
n & \quad \mid\ A \wedge (B \vee C) & \wedge\mathrm{I}\ i, n-1 \\
n+1 & \quad \mid\ A \wedge C & \\
 & \quad \mid\ \vdots & \\
m & \quad \mid\ A \wedge (B \vee C) & \\
m+1 & A \wedge (B \vee C) & \vee\mathrm{E}\ 1, 2\text{–}n, (n+1)\text{–}m
\end{array}$$

We immediately see that we get line i from 2 by $\wedge$E; let's apply the strategy for proving disjunctions to line $n-1$: look for a proof of B. (We have a choice of which disjunct to pick, but picking C wouldn't work and we'd end up having to backtrack.)

$$\begin{array}{lll}
1 & (A \wedge B) \vee (A \wedge C) & \\
\hline
2 & \quad \mid\ A \wedge B & \\
3 & \quad \mid\ A & \wedge\mathrm{E}\ 2 \\
4 & \quad \mid\ B & \wedge\mathrm{E}\ 2 \\
5 & \quad \mid\ B \vee C & \vee\mathrm{I}\ 4 \\
6 & \quad \mid\ A \wedge (B \vee C) & \wedge\mathrm{I}\ 3, 5 \\
7 & \quad \mid\ A \wedge C & \\
 & \quad \mid\ \vdots & \\
m & \quad \mid\ A \wedge (B \vee C) & \\
m+1 & A \wedge (B \vee C) & \vee\mathrm{E}\ 1, 2\text{–}6, 7\text{–}m
\end{array}$$

Like line i (i.e., 3), we get line 4 from 2 by $\wedge$E. That's it for the first subproof. The second subproof is almost exactly the same. We'll leave it as an exercise.

Remember that when we started, we had the option of working forward from the premise, or working backward from the conclusion, and we picked the first option. The second option also leads to a proof, but it will look different. The first steps would be to split the conclusion apart and set up two subgoals, A and $B \vee C$, and then work forwards from the premise to prove them, e.g.,:

$$\begin{array}{lll}
1 & \mid\ \underline{(A \wedge B) \vee (A \wedge C)} & \\
2 & \mid\ \mid\ \underline{A \wedge B} & \\
 & \mid\ \mid\ \vdots & \\
k & \mid\ \mid\ A & \\
k+1 & \mid\ \mid\ \underline{A \wedge C} & \\
 & \mid\ \mid\ \vdots & \\
n-1 & \mid\ \mid\ A & \\
n & \mid\ A & \vee\text{E } 1, 2\text{–}k, (k+1)\text{–}(n-1) \\
n+1 & \mid\ \mid\ \underline{A \wedge B} & \\
 & \mid\ \mid\ \vdots & \\
l & \mid\ \mid\ B \vee C & \\
l+1 & \mid\ \mid\ \underline{A \wedge C} & \\
 & \mid\ \mid\ \vdots & \\
m-1 & \mid\ \mid\ B \vee C & \\
m & \mid\ B \vee C & \vee\text{E } 1, (n+1)\text{–}l, (l+1)\text{–}(m-1) \\
m+1 & \mid\ A \wedge (B \vee C) & \wedge\text{I } n, m-1
\end{array}$$

Let's give another example to illustrate how to apply the strategies to deal with conditionals and negation. The sentence $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$ is a tautology; let's see if we can find a proof of it, from no premises, using the strategies. We first write the sentence at the bootom of a sheet of paper. Since working forwards is not an option, we work backwards, and set up a subproof to establish the sentence we want using $\rightarrow$I. Its assumption must be the antecedent of the conditional we want to prove, and its last line the consequent.

$$\begin{array}{lll}
1 & \mid\ \mid\ \underline{A \rightarrow B} & \\
 & \mid\ \mid\ \vdots & \\
n & \mid\ \mid\ \neg B \rightarrow \neg A & \\
n+1 & \mid\ (A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A) & \rightarrow\text{I } 1\text{–}n
\end{array}$$

The new goal, $\neg B \rightarrow \neg A$ is itself a conditional, so working backwards we set up another subproof:

```
1     │ │ A → B
      │ ├──────
2     │ │ │ ¬B
      │ │ ├────
      │ │ │ ⋮
n − 1 │ │ │ ¬A
n     │ │ ¬B → ¬A                     →I 2–(n − 1)
n + 1 │ (A → B) → (¬B → ¬A)           →I 1–n
```

From $\neg A$ we again work backwards:

```
1     │ │ A → B
      │ ├──────
2     │ │ │ ¬B
      │ │ ├────
3     │ │ │ │ A
      │ │ │ ├───
      │ │ │ │ ⋮
n − 2 │ │ │ │ ⊥
n − 1 │ │ │ ¬A                        ¬I 3–(n − 2)
n     │ │ ¬B → ¬A                     →I 2–(n − 1)
n + 1 │ (A → B) → (¬B → ¬A)           →I 1–n
```

To prove $\bot$ we now work forwards from the one negated sentence we have, $\neg B$ on line 2. That means we have to derive B—which we get in turn by working forwards from $A \rightarrow B$, since B is the consequent of that conditional. Its antecedent A is already available on line 3. So we finish with:

```
1 │ │ A → B
  │ ├──────
2 │ │ │ ¬B
  │ │ ├────
3 │ │ │ │ A
  │ │ │ ├───
4 │ │ │ │ B                 →E 1, 3
5 │ │ │ │ ⊥                 ⊥I 2, 4
6 │ │ │ ¬A                  ¬I 3–5
7 │ │ ¬B → ¬A               →I 2–6
8 │ (A → B) → (¬B → ¬A)     →I 1–7
```

16.4 Use Negation Elimination

In very many cases, the strategies of working forwards and backwards will eventually pan out. But there are cases where they do not work. If you cannot find a way to show $\mathscr{A}$ directly using those, use ¬E instead. To do this, set up a subproof in which you assume $\neg\mathscr{A}$ and look for a proof of ⊥ inside that subproof.

```
        │ ⋮
n       │ │ ¬A
        │ ├────
        │ │ ⋮
m       │ │ ⊥
m + 1   │ 𝒜          ¬E n–m
```

Here, we have to start a subproof with assumption $\neg\mathscr{A}$; the last line of the subproof has to be ⊥. We'll cite the subproof, and use ¬E. In the subproof, we now have an additional assumption (on line n) to work with.

Suppose we used the ¬E proof strategy, or we're in some other situation where we're looking for a proof of ⊥. What's a good candidate? Of course the obvious candidate would be to use a negated sentence, since (as we saw above) ⊥I always yields ⊥. If you are using the ¬E strategy, this results in the following paradoxical situation:

```
        │ ⋮
n       │ │ ¬𝒜
        │ ├────
        │ │ ⋮
m − 1   │ │ 𝒜
m       │ │ ⊥        ⊥I n, m − 1
m + 1   │ 𝒜          ¬E n–m
```

This looks weird: We wanted to prove $\mathscr{A}$ and the strategies failed us; so we used ¬E as a last resort. And now we find ourselves in the same situation: looking for a proof of $\mathscr{A}$. But remember that we are now inside a subproof, and in that subproof we have an additional assumption ($\neg\mathscr{A}$) to work with which we didn't have before.

16.5 A proof of excluded middle

The sentence $A \lor \neg A$ is a tautology, and so should have a proof even without any premises. But working backwards does not work: to get $A \lor \neg A$ using ∨I we would have to prove either A or $\neg A$—again, from no premises. But neither of these is a tautology, so we won't be able to prove either. Working forwards doesn't work either, since there is nothing to work forwards from. So, the only option is ¬E.

```
1     │ │ ¬(A ∨ ¬A)
      │ ├──────────
      │ │ ⋮
m     │ │ ⊥
m + 1 │ A ∨ ¬A          ¬E 1–m
```

Now we do have something to work forward from: the assumption $\neg(A \vee \neg A)$. To use it, we justify $\bot$ by $\bot$I, citing the assumption, in addition to the corresponding unnegated sentence (yet to be proved).

```
1     │ │ ¬(A ∨ ¬A)
      │ ├──────────
      │ │ ⋮
m − 1 │ │ A ∨ ¬A
m     │ │ ⊥               ⊥I 1, m − 1
m + 1 │ A ∨ ¬A          ¬E 1–m
```

At the outset, working backwards to prove $A \vee \neg A$ by $\vee$I did not work. But we are now in a different situation: we want to prove $A \vee \neg A$ inside a subproof. In general, when dealing with new goals we should go back and start with the basic strategies. In this case, we should pick a disjunct and try to prove it. Let's pick $\neg A$. Then working backward from that, we start another subproof in order to justify $\neg A$ using $\neg$I.

```
1     │ │ ¬(A ∨ ¬A)
      │ ├──────────
2     │ │ │ A
      │ │ ├───
      │ │ │ ⋮
m − 3 │ │ │ ⊥
m − 2 │ │ ¬A              ¬I 2–(m − 3)
m − 1 │ │ A ∨ ¬A          ∨I m − 2
m     │ │ ⊥               ⊥I 1, m − 1
m + 1 │ A ∨ ¬A          ¬I 1–m
```

Inside this new subproof, we need to justify $\bot$. Again, the best way to do this is to use a negated sentence, and $\neg(A \vee \neg A)$ is the only negated sentence we can use. The corresponding unnegated sentence, $A \vee \neg A$, however, directly follows from A by $\vee$I. Our complete proof is:

$$
\begin{array}{r|l l}
1 & \quad \mid \ \underline{\neg(A \vee \neg A)} & \\
2 & \quad \mid \quad \mid \ \underline{A} & \\
3 & \quad \mid \quad \mid \ A \vee \neg A & \vee\mathrm{I}\ 2 \\
4 & \quad \mid \quad \mid \ \bot & \bot\mathrm{I}\ 1, 3 \\
5 & \quad \mid \ \neg A & \neg\mathrm{I}\ 2\text{–}4 \\
6 & \quad \mid \ A \vee \neg A & \vee\mathrm{I}\ 5 \\
7 & \quad \mid \ \bot & \bot\mathrm{I}\ 1, 6 \\
8 & A \vee \neg A & \neg\mathrm{I}\ 1\text{–}7
\end{array}
$$

17 | Additional rules for SL

In §15, we introduced the basic rules of our proof system for SL. In this section, we will add some additional rules to our system. These will make our system much easier to work with. (However, in §19 we will see that they are not strictly speaking *necessary*.)

17.1 Reiteration

The first additional rule is *reiteration* (R). This just allows us to repeat ourselves:

$$
\begin{array}{r|ll}
m & \mathscr{A} & \\
 & \mathscr{A} & \text{R } m
\end{array}
$$

Such a rule is obviously legitimate; but one might well wonder how such a rule could ever be useful. Well, consider:

$$
\begin{array}{r|lll}
1 & A \to \neg A & & \\ \hline
2 & \quad | \ A & & \\
3 & \quad | \ \neg A & & \to\text{E } 1, 2 \\
4 & \quad | \ \neg A & & \\
5 & \quad | \ \neg A & & \text{R } 4 \\
6 & \neg A & & \text{LEM } 2\text{–}3, 4\text{–}5
\end{array}
$$

This is a fairly typical use of the R rule.

17.2 Disjunctive syllogism

Here is a very natural argument form.

> Elizabeth is either in Massachusetts or in DC. She is not in DC. So, she is in Massachusetts.

This inference pattern is called *disjunctive syllogism*. We add it to our proof system as follows:

$$\begin{array}{r|ll} m & \mathscr{A} \lor \mathscr{B} & \\ n & \neg\mathscr{A} & \\ & \mathscr{B} & \text{DS } m, n \end{array}$$

and

$$\begin{array}{r|ll} m & \mathscr{A} \lor \mathscr{B} & \\ n & \neg\mathscr{B} & \\ & \mathscr{A} & \text{DS } m, n \end{array}$$

As usual, the disjunction and the negation of one disjunct may occur in either order and need not be adjacent. However, we always cite the disjunction first.

17.3 Modus tollens

Another useful pattern of inference is embodied in the following argument:

> If Mitt has won the election, then he is in the White House. He is not in the White House. So he has not won the election.

This inference pattern is called *modus tollens*. The corresponding rule is:

$$\begin{array}{r|ll} m & \mathscr{A} \to \mathscr{B} & \\ n & \neg\mathscr{B} & \\ & \neg\mathscr{A} & \text{MT } m, n \end{array}$$

As usual, the premises may occur in either order, but we always cite the conditional first.

17.4 Double-negation elimination

Another useful rule is *double-negation elimination*. This rule does exactly what it says on the tin:

$$\begin{array}{r|ll} m & \neg\neg\mathscr{A} & \\ & \mathscr{A} & \text{DNE } m \end{array}$$

The justification for this is that, in natural language, double-negations tend to cancel out.

That said, you should be aware that context and emphasis can prevent them from doing so. Consider: 'Jane is not *not* happy'. Arguably, one cannot infer 'Jane is happy', since the first sentence should be understood as meaning the same as 'Jane is not *un*happy'. This is compatible with 'Jane is in a state of profound indifference'. As usual, moving to SL forces us to sacrifice certain nuances of English expressions.

17.5 Excluded middle

Suppose that we can show that if it's sunny outside, then Bill will have brought an umbrella (for fear of burning). Suppose we can also show that, if it's not sunny outside, then Bill will have brought an umbrella (for fear of rain). Well, there is no third way for the weather to be. So, *whatever the weather*, Bill will have brought an umbrella.

This line of thinking motivates the following rule:

$$\begin{array}{r|lll}
i & \vert & \underline{\mathscr{A}} & \\
j & \vert & \mathscr{B} & \\
k & \vert & \underline{\neg\mathscr{A}} & \\
l & \vert & \mathscr{B} & \\
 & \mathscr{B} & & \text{LEM } i\text{–}j,\ k\text{–}l
\end{array}$$

The rule is sometimes called the law of *excluded middle*, since it encapsulates the idea that $\mathscr{A}$ can be true or $\neg\mathscr{A}$ may be true, but there is no middle way where neither is true.[1] There can be as many lines as you like between i and j, and as many lines as you like between k and l. Moreover, the subproofs can come in any order, and the second subproof does not need to come immediately after the first.

To see the rule in action, consider:

$$P \therefore (P \wedge D) \vee (P \wedge \neg D)$$

Here is a proof corresponding with the argument:

[1] You may sometimes find logicians or philosophers talking about "tertium non datur." That's the same principle as excluded middle; it means "no third way." Logicians who have qualms about ¬E also have qualms about LEM.

$$
\begin{array}{rlll}
1 & \mid \underline{P} & & \\
2 & \mid \quad \mid \underline{D} & & \\
3 & \mid \quad \mid P \wedge D & & \wedge\text{I } 1, 2 \\
4 & \mid \quad \mid (P \wedge D) \vee (P \wedge \neg D) & & \vee\text{I } 3 \\
5 & \mid \quad \mid \underline{\neg D} & & \\
6 & \mid \quad \mid P \wedge \neg D & & \wedge\text{I } 1, 5 \\
7 & \mid \quad \mid (P \wedge D) \vee (P \wedge \neg D) & & \vee\text{I } 6 \\
8 & \mid (P \wedge D) \vee (P \wedge \neg D) & & \text{LEM } 2\text{–}4, 5\text{–}7
\end{array}
$$

17.6 De Morgan Rules

Our final additional rules are called De Morgan's Laws (named after Augustus De Morgan). The shape of the rules should be familiar from truth tables.

The first De Morgan rule is:

$$
\begin{array}{rlll}
m & \mid \neg(\mathscr{A} \wedge \mathscr{B}) & & \\
 & \mid \neg\mathscr{A} \vee \neg\mathscr{B} & & \text{DeM } m
\end{array}
$$

The second De Morgan is the reverse of the first:

$$
\begin{array}{rlll}
m & \mid \neg\mathscr{A} \vee \neg\mathscr{B} & & \\
 & \mid \neg(\mathscr{A} \wedge \mathscr{B}) & & \text{DeM } m
\end{array}
$$

The third De Morgan rule is the *dual* of the first:

$$
\begin{array}{rlll}
m & \mid \neg(\mathscr{A} \vee \mathscr{B}) & & \\
 & \mid \neg\mathscr{A} \wedge \neg\mathscr{B} & & \text{DeM } m
\end{array}
$$

And the fourth is the reverse of the third:

$$
\begin{array}{rlll}
m & \mid \neg\mathscr{A} \wedge \neg\mathscr{B} & & \\
 & \mid \neg(\mathscr{A} \vee \mathscr{B}) & & \text{DeM } m
\end{array}
$$

These are all of the additional rules of our proof system for SL.

Practice exercises

A. The following proofs are missing their citations (rule and line numbers). Add them wherever they are required:

```
1  | W → ¬B
2  | A ∧ W
3  | B ∨ (J ∧ K)
   |------------
4  | W
5  | ¬B
6  | J ∧ K
7  | K
```

```
1  | L ↔ ¬O
2  | L ∨ ¬O
   |--------
3  | | ¬L
   | |----
4  | | ¬O
5  | | L
6  | | ⊥
7  | ¬¬L
8  | L
```

```
1  | Z → (C ∧ ¬N)
2  | ¬Z → (N ∧ ¬C)
   |--------------
3  | | ¬(N ∨ C)
   | |---------
4  | | ¬N ∧ ¬C
5  | | ¬N
6  | | ¬C
7  | | | Z
   | | |----
8  | | | C ∧ ¬N
9  | | | C
10 | | | ⊥
11 | | ¬Z
12 | | N ∧ ¬C
13 | | N
14 | | ⊥
15 | ¬¬(N ∨ C)
16 | N ∨ C
```

B. Give a proof for each of these arguments:

1. $E \vee F, F \vee G, \neg F \therefore E \wedge G$
2. $M \vee (N \rightarrow M) \therefore \neg M \rightarrow \neg N$
3. $(M \vee N) \wedge (O \vee P), N \rightarrow P, \neg P \therefore M \wedge O$
4. $(X \wedge Y) \vee (X \wedge Z), \neg(X \wedge D), D \vee M \therefore M$

18 | Proof-theoretic concepts

In this chapter we will introduce some new vocabulary. The following expression:

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vdash \mathscr{C}$$

means that there is some proof which starts with assumptions among $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ and ends with $\mathscr{C}$ (and contains no undischarged assumptions other than those we started with). Derivatively, we will write:

$$\vdash \mathscr{A}$$

to mean that there is a proof of $\mathscr{A}$ with no assumptions.

The symbol '$\vdash$' is called the *single turnstile*. We want to emphasize that this is not the double turnstile symbol ('$\vDash$') that we introduced in chapter 11 to symbolize entailment. The single turnstile, '$\vdash$', concerns the existence of proofs; the double turnstile, '$\vDash$', concerns the existence of valuations (or interpretations, when used for FOL). *They are very different notions.*

Armed with our '$\vdash$' symbol, we can introduce some more terminology. To say that there is a proof of $\mathscr{A}$ with no undischarged assumptions, we write: $\vdash \mathscr{A}$. In this case, we say that $\mathscr{A}$ is a THEOREM.

$\mathscr{A}$ is a THEOREM iff $\vdash \mathscr{A}$

To illustrate this, suppose we want to show that '$\neg(A \land \neg A)$' is a theorem. So we need a proof of '$\neg(A \land \neg A)$' which has *no* undischarged assumptions. However, since we want to prove a sentence whose main operator is a negation, we will want to start with a *subproof* within which we assume '$A \land \neg A$', and show that this assumption leads to contradiction. All told, then, the proof looks like this:

$$\begin{array}{r|l|ll}
1 & & \underline{A \land \neg A} & \\
2 & & A & \land\text{E } 1 \\
3 & & \neg A & \land\text{E } 1 \\
4 & & \bot & \bot\text{I } 3, 2 \\
5 & \multicolumn{2}{l}{\neg(A \land \neg A)} & \neg\text{I } 1\text{–}4
\end{array}$$

We have therefore proved '$\neg(A \wedge \neg A)$' on no (undischarged) assumptions. This particular theorem is an instance of what is sometimes called *the Law of Non-Contradiction*.

To show that something is a theorem, you just have to find a suitable proof. It is typically much harder to show that something is *not* a theorem. To do 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. Perhaps you just didn't try hard enough.

Here is another new bit of terminology:

Two sentences $\mathscr{A}$ and $\mathscr{B}$ are PROVABLY EQUIVALENT iff each can be proved from the other; i.e., both $\mathscr{A} \vdash \mathscr{B}$ and $\mathscr{B} \vdash \mathscr{A}$.

As in the case of showing that a sentence is a theorem, 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 is just as hard as showing that a sentence is not a theorem.

Here is a third, related, bit of terminology:

The sentences $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are PROVABLY INCONSISTENT iff a contradiction can be proved from them, i.e. $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vdash \bot$. If they are not INCONSISTENT, we call them PROVABLY CONSISTENT.

It is easy to show that some sentences are provably inconsistent: you just need to prove a contradiction from assuming all the sentences. Showing that some sentences are not provably inconsistent is much harder. It would require more than just providing a proof or two; it would require showing that no proof of a certain kind is *possible*.

This table summarises whether one or two proofs suffice, or whether we must reason about all possible proofs.

	Yes	**No**
theorem?	one proof	all possible proofs
inconsistent?	one proof	all possible proofs
equivalent?	two proofs	all possible proofs
consistent?	all possible proofs	one proof

Practice exercises

A. Show that each of the following sentences is a theorem:

1. $O \rightarrow O$
2. $N \vee \neg N$

3. $J \leftrightarrow [J \vee (L \wedge \neg L)]$
4. $((A \rightarrow B) \rightarrow A) \rightarrow A$

B. Provide proofs to show each of the following:

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

C. Show that each of the following pairs of sentences are provably equivalent:

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

D. If you know that $\mathscr{A} \vdash \mathscr{B}$, what can you say about $(\mathscr{A} \wedge \mathscr{C}) \vdash \mathscr{B}$? What about $(\mathscr{A} \vee \mathscr{C}) \vdash \mathscr{B}$? Explain your answers.

E. In this chapter, we claimed that it is just as hard to show that two sentences are not provably equivalent, as it is to show that a sentence is not a theorem. Why did we claim this? (*Hint*: think of a sentence that would be a theorem iff $\mathscr{A}$ and $\mathscr{B}$ were provably equivalent.)

19 Derived rules

In this section, we will see why we introduced the rules of our proof system in two separate batches. In particular, we want to show that the additional rules of §17 are not strictly speaking necessary, but can be derived from the basic rules of §15.

19.1 Derivation of Reiteration

Suppose you have some sentence on some line of your deduction:

m	$\mathscr{A}$

You now want to repeat yourself, on some line k. You could just invoke the rule R, introduced in §17. But equally well, you can do this with the *basic* rules of §15:

m	$\mathscr{A}$	
k	$\mathscr{A} \wedge \mathscr{A}$	$\wedge$I m, m
$k+1$	$\mathscr{A}$	$\wedge$E k

To be clear: this is not a proof. Rather, it is a proof *scheme*. After all, it uses a variable, '$\mathscr{A}$', rather than a sentence of SL, but the point is simple: Whatever sentences of SL we plugged in for '$\mathscr{A}$', and whatever lines we were working on, we could produce a bona fide proof. So you can think of this as a recipe for producing proofs.

Indeed, it is a recipe which shows us that, anything we can prove using the rule R, we can prove (with one more line) using just the *basic* rules of §15. So we can describe the rule R as a *derived* rule, since it can be justified using only the basic rules.

19.2 Derivation of Disjunctive Syllogism

Suppose that you are in a proof, and you have something of this form:

m	$\mathscr{A} \vee \mathscr{B}$
n	$\neg\mathscr{A}$

You now want, on line k, to prove $\mathscr{B}$. You can do this with the rule of DS, introduced in §17, but equally well, you can do this with the *basic* rules of §15:

$$\begin{array}{l|lll}
m & \mathscr{A} \lor \mathscr{B} & & \\
n & \neg\mathscr{A} & & \\
k & \quad\vert\ \underline{\mathscr{A}} & & \\
k+1 & \quad\vert\ \bot & & \bot\text{I } n, k \\
k+2 & \quad\vert\ \mathscr{B} & & \bot\text{E } k+1 \\
k+3 & \quad\vert\ \underline{\mathscr{B}} & & \\
k+4 & \quad\vert\ \mathscr{B} \land \mathscr{B} & & \land\text{I } k+3, k+3 \\
k+5 & \quad\vert\ \mathscr{B} & & \land\text{E } k+4 \\
k+6 & \mathscr{B} & & \lor\text{E } m, k\text{–}k+2, k+3\text{–}k+5
\end{array}$$

So the DS rule, again, can be derived from our more basic rules. Adding it to our system did not make any new proofs possible. Anytime you use the DS rule, you could always take a few extra lines and prove the same thing using only our basic rules. It is a *derived* rule.

19.3 Derivation of Modus tollens

Suppose you have the following in your proof:

$$\begin{array}{l|l}
m & \mathscr{A} \to \mathscr{B} \\
n & \neg\mathscr{B}
\end{array}$$

You now want, on line k, to prove $\neg\mathscr{A}$. You can do this with the rule of MT, introduced in §17. Equally well, you can do this with the *basic* rules of §15:

$$\begin{array}{l|lll}
m & \mathscr{A} \to \mathscr{B} & & \\
n & \neg\mathscr{B} & & \\
k & \quad\vert\ \underline{\mathscr{A}} & & \\
k+1 & \quad\vert\ \mathscr{B} & & \to\text{E } m, k \\
k+2 & \quad\vert\ \bot & & \bot\text{I } n, k+1 \\
k+3 & \neg\mathscr{A} & & \neg\text{I } k\text{–}k+2
\end{array}$$

Again, the rule of MT can be derived from the *basic* rules of §15.

19.4 Derivation of Double-negation elimination

Consider the following deduction scheme:

$$\begin{array}{l|l l}
m & \neg\neg\mathscr{A} & \\
k & \quad\vert\ \underline{\neg\mathscr{A}} & \\
k+1 & \quad\vert\ \bot & \bot\text{I } m, k \\
k+2 & \mathscr{A} & \neg\text{E } k\text{–}k+1
\end{array}$$

Again, we can derive the DNE rule from the *basic* rules of §15.

19.5 Derivation of excluded middle

Suppose you want to prove something using the LEM rule, i.e., you have in your proof

$$\begin{array}{l|l}
m & \quad\vert\ \underline{\mathscr{A}} \\
n & \quad\vert\ \mathscr{B} \\
k & \quad\vert\ \underline{\neg\mathscr{A}} \\
l & \quad\vert\ \mathscr{B}
\end{array}$$

You now want, on line $l + 1$, to prove $\mathscr{B}$. The rule LEM from §17 would allow you to do it. But can do this with the *basic* rules of §15?

One option is to first prove $\mathscr{A} \lor \neg\mathscr{A}$, and then apply $\lor$E, i.e. proof by cases:

$$\begin{array}{l|l l}
m & \quad\vert\ \underline{\mathscr{A}} & \\
n & \quad\vert\ \mathscr{B} & \\
k & \quad\vert\ \underline{\neg\mathscr{A}} & \\
l & \quad\vert\ \mathscr{B} & \\
 & \dots & \\
i & \mathscr{A} \lor \neg\mathscr{A} & \\
i+1 & \mathscr{B} & \lor\text{E } i, m\text{–}n, k\text{–}l
\end{array}$$

(We gave a proof of $\mathscr{A} \lor \neg\mathscr{A}$ using only our basic rules in §16.5.)

Here is another way that is a bit more complicated than the ones before. What you have to do is embed your two subproofs inside another subproof. The assumption of the subproof will be $\neg\mathscr{B}$, and the last line will be $\bot$. Thus, the complete subproof is the kind you need to conclude $\mathscr{B}$ using ¬E. Inside the proof, you'd have to do a bit more work to get ⊥:

```
m      │ │ ¬𝓑
       │ ├────
m + 1  │ │ │ 𝓐
       │ │ ├────
n + 1  │ │ │ 𝓑
n + 2  │ │ │ ⊥        ⊥I m, n + 1
k + 2  │ │ │ ¬𝓐
       │ │ ├────
l + 2  │ │ │ 𝓑
l + 3  │ │ │ ⊥        ⊥I k + 2, l + 2
l + 4  │ │ ¬𝓐         ¬I m + 1–n + 2
l + 5  │ │ ¬¬𝓐        ¬I k + 2–l + 3
l + 6  │ │ ⊥          ⊥I l + 5, l + 4
l + 7  │ 𝓑            ¬E m–l + 6
```

Note that because we add an assumption at the top and additional conclusions inside the subproofs, the line numbers change. You may have to stare at this for a while before you understand what's going on.

19.6 Derivation of De Morgan rules

Here is a demonstration of how we could derive the first De Morgan rule:

```
m      │ ¬(𝓐 ∧ 𝓑)
k      │ │ 𝓐
       │ ├────
k + 1  │ │ │ 𝓑
       │ │ ├────
k + 2  │ │ │ 𝓐 ∧ 𝓑     ∧I k, k + 1
k + 3  │ │ │ ⊥         ⊥I m, k + 2
k + 4  │ │ ¬𝓑          ¬I k + 1–k + 3
k + 5  │ │ ¬𝓐 ∨ ¬𝓑     ∨I k + 4
k + 6  │ │ ¬𝓐
       │ ├────
k + 7  │ │ ¬𝓐 ∨ ¬𝓑     ∨I k + 6
k + 8  │ ¬𝓐 ∨ ¬𝓑       LEM k–k + 5, k + 6–k + 7
```

Here is a demonstration of how we could derive the second De Morgan rule:

$$
\begin{array}{l|l l}
m & \neg\mathscr{A} \lor \neg\mathscr{B} & \\
k & \quad\vert\ \underline{\mathscr{A} \land \mathscr{B}} & \\
k+1 & \quad\vert\ \mathscr{A} & \land\text{E } k \\
k+2 & \quad\vert\ \mathscr{B} & \land\text{E } k \\
k+3 & \quad\vert\quad\vert\ \underline{\neg\mathscr{A}} & \\
k+4 & \quad\vert\quad\vert\ \bot & \bot\text{I } k+3, k+1 \\
k+5 & \quad\vert\quad\vert\ \underline{\neg\mathscr{B}} & \\
k+6 & \quad\vert\quad\vert\ \bot & \bot\text{I } k+5, k+2 \\
k+7 & \quad\vert\ \bot & \lor\text{E } m, k+3\text{–}k+4, k+5\text{–}k+6 \\
k+8 & \neg(\mathscr{A} \land \mathscr{B}) & \neg\text{I } k\text{–}k+7
\end{array}
$$

Similar demonstrations can be offered explaining how we could derive the third and fourth De Morgan rules. These are left as exercises.

Practice exercises

A. Provide proof schemes that justify the addition of the third and fourth De Morgan rules as derived rules.

B. The proofs you offered in response to the practice exercises of §§17–18 used derived rules. Replace the use of derived rules, in such proofs, with only basic rules. You will find some 'repetition' in the resulting proofs; in such cases, offer a streamlined proof using only basic rules. (This will give you a sense, both of the power of derived rules, and of how all the rules interact.)

C. Give a proof of $\mathscr{A} \lor \neg\mathscr{A}$. Then give a proof that *uses only the basic rules*.

D. Show that if you had LEM as a basic rule, you could justify ¬E as a derived rule. That is, suppose you had the proof:

$$
\begin{array}{l|l}
m & \quad\vert\ \underline{\neg\mathscr{A}} \\
 & \quad\vert\ \dots \\
n & \quad\vert\ \bot
\end{array}
$$

How could you use it to prove $\mathscr{A}$ without using ¬E but with using LEM as well as all the other primitive rules?

E. Give a proof of the first De Morgan rule, but using only the basic rules, in particular, *without using LEM*. (Of course, you can combine the proof using LEM with the proof *of* LEM. Try to find a proof directly.)

20 | Soundness and completeness

In §18, we saw that we could use derivations to test for the same concepts we used truth tables to test for. Not only could we use derivations to prove that an argument is valid, we could also use them to test if a sentence is a tautology or a pair of sentences are equivalent. We also started using the single turnstile the same way we used the double turnstile. If we could prove that $\mathscr{A}$ was a tautology with a truth table, we wrote $\vDash \mathscr{A}$, and if we could prove it using a derivation, we wrote $\vdash \mathscr{A}$.

You may have wondered at that point if the two kinds of turnstiles always worked the same way. If you can show that $\mathscr{A}$ is a tautology using truth tables, can you also always show that it is true using a derivation? Is the reverse true? Are these things also true for tautologies and pairs of equivalent sentences? As it turns out, the answer to all these questions and many more like them is yes. We can show this by defining all these concepts separately and then proving them equivalent. That is, we imagine that we actually have two notions of validity, valid$_{\vDash}$ and valid$_{\vdash}$ and then show that the two concepts always work the same way.

To begin with, we need to define all of our logical concepts separately for truth tables and derivations. A lot of this work has already been done. We handled all of the truth table definitions in §11. We have also already given syntactic definitions for tautologies (theorems) and pairs of logically equivalent sentences. The other definitions follow naturally. For most logical properties we can devise a test using derivations, and those that we cannot test for directly can be defined in terms of the concepts that we can define.

For instance, we defined a theorem as a sentence that can be derived without any premises (p. 109). Since the negation of a contradiction is a tautology, we can define a SYNTACTIC CONTRADICTION IN SL as a sentence whose negation can be derived without any premises. The syntactic definition of a contingent sentence is a little different. We don't have any practical, finite method for proving that a sentence is contingent using derivations, the way we did using truth tables. So we have to content ourselves with defining "contingent sentence" negatively. A sentence is SYNTACTICALLY CONTINGENT IN SL if it is not a theorem or a contradiction.

A collection of sentences are PROVABLY INCONSISTENT IN SL if and only if one can derive a contradiction from them. Consistency, on the other hand, is like contingency, in that we do not have a practical finite method to test for it directly. So again, we have to define a term negatively. A collection of sentences is PROVABLY CONSISTENT IN SL if and only if they are not provably inconsistent.

Finally, an argument is PROVABLY VALID IN SL if and only if there is a derivation of its conclusion from its premises. All of these definitions are given in Table 20.1.

All of our concepts have now been defined both semantically and syntactically. How can we prove that these definitions always work the same way? A full proof here goes well beyond the scope of this book. However, we can sketch what it would be like. We will focus on showing the two notions of validity to be equivalent. From that the other concepts will follow quickly. The proof will have to go in two directions. First we will have to show that things which are syntactically valid will also be semantically valid. In other words, everything that we can prove using derivations could also be proven using truth tables. Put symbolically, we want to show that valid$_{\vdash}$ implies valid$_{\vDash}$. Afterwards, we will need to show things in the other directions, valid$_{\vDash}$ implies valid$_{\vdash}$

This argument from $\vdash$ to $\vDash$ is the problem of SOUNDNESS. A proof system is SOUND if there are no derivations of arguments that can be shown invalid by truth tables. 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.

The proof that we will sketch depends on the fact that we initially defined a sentence of SL using a recursive definition (see p. 32). We could have also used recursive definitions to define a proper proof in SL and a proper truth table. (Although we didn't.) If we had these definitions, we could then use a *recursive proof* to show the soundness of SL. A recursive proof works the same way as a recursive definition. With the recursive definition, we identified a group of base elements that were stipulated to be examples of the thing we were trying to define. In the case of a SL sentence, the base class was the set of sentence letters A, B, C, ... We just announced that these were sentences. The second step of a recursive definition is to say that anything that is built up from your base class using certain rules also counts as an example of the thing you are defining. In the case of a definition of a sentence, the rules corresponded to the five sentential operators (see p. 32). Once you have established a recursive definition, you can use that definition to show that all the members of the class you have defined have a certain property. You simply prove that the property is true of the members of the base class, and then you prove that the rules for extending the base class don't change the property. This is what it means to give a recursive proof.

Even though we don't have a recursive definition of a proof in SL, we can sketch how a recursive proof of the soundness of SL would go. Imagine a base class of one-line proofs, one for each of our eleven rules of inference. The members of this class would look like this $\mathcal{A}, \mathcal{B} \vdash \mathcal{A} \wedge \mathcal{B}$; $\mathcal{A} \wedge \mathcal{B} \vdash \mathcal{A}$; $\mathcal{A} \vee \mathcal{B}, \neg\mathcal{A} \vdash \mathcal{B}$ ... etc. Since some rules have a couple different forms, we would have to have add some members to this base class, for instance $\mathcal{A} \wedge \mathcal{B} \vdash \mathcal{B}$ Notice that these are all statements in the metalanguage. The proof that SL is sound is not a part of SL, because SL does not have the power to talk about itself.

You can use truth tables to prove to yourself that each of these one-line proofs in this base class is valid$_{\vDash}$. For instance the proof $\mathcal{A}, \mathcal{B} \vdash \mathcal{A} \wedge \mathcal{B}$ corresponds to a truth table that shows $\mathcal{A}, \mathcal{B} \vDash \mathcal{A} \wedge \mathcal{B}$ This establishes the first part of our recursive proof.

The next step is to show that adding lines to any proof will never change a valid$_{\vDash}$

Concept	Truth table (semantic) definition	Proof-theoretic (syntactic) definition
Tautology	A sentence whose truth table only has Ts under the main operator	A sentence that can be derived without any premises.
Contradiction	A sentence whose truth table only has Fs under the main operator	A sentence whose negation can be derived without any premises
Contingent sentence	A sentence whose truth table contains both Ts and Fs under the main operator	A sentence that is not a theorem or contradiction
Equivalent sentences	The columns under the main operators are identical.	The sentences can be derived from each other
Unsatisfiable/ inconsistent sentences	Sentences which do not have a single line in their truth table where they are all true.	Sentences from which one can derive a contradiction
Satisfiable/ Consistent sentences	Sentences which have at least one line in their truth table where they are all true.	Sentences from which one cannot derive a contradiction
Valid argument	An argument whose truth table has no lines where there are all Ts under main operators for the premises and an F under the main operator for the conclusion.	An argument where one can derive the conclusion from the premises

Table 20.1: Two ways to define logical concepts.

proof into an invalid$_{\vDash}$ one. We would need to do this for each of our eleven basic rules of inference. So, for instance, for ∧I we need to show that for any proof $\mathscr{A}_1, \ldots, \mathscr{A}_n \vdash \mathscr{B}$ adding a line where we use ∧I to infer $\mathscr{C} \wedge \mathscr{D}$, where $\mathscr{C} \wedge \mathscr{D}$ can be legitimately inferred from $\mathscr{A}_1, \ldots, \mathscr{A}_n, \mathscr{B}$, would not change a valid proof into an invalid proof. But wait, if we can legitimately derive $\mathscr{C} \wedge \mathscr{D}$ from these premises, then $\mathscr{C}$ and $\mathscr{D}$ must be already available in the proof. They are either already among $\mathscr{A}_1, \ldots, \mathscr{A}_n, \mathscr{B}$, or can be legitimately derived from them. As such, any truth table line in which the premises are true must be a truth table line in which $\mathscr{C}$ and $\mathscr{D}$ are true. According to the characteristic truth table for ∧, this means that $\mathscr{C} \wedge \mathscr{D}$ is also true on that line. Therefore, $\mathscr{C} \wedge \mathscr{D}$ validly follows from the premises. This means that using the ∧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 11 other basic rules. This tedious exercise falls beyond the scope of this book.

So we have shown that $\mathscr{A} \vdash \mathscr{B}$ implies $\mathscr{A} \vDash \mathscr{B}$. What about the other direction, that is why think that *every* argument that can be shown valid using truth tables can also be proven using a derivation.

This is the problem of completeness. A proof system has the property of COMPLETENESS if and only if there is a derivation of every semantically valid argument. Proving that a system is complete is generally harder than proving that it is sound. Proving that a system is sound amounts to showing that all of the rules of your proof system work the way they are supposed to. Showing that a system is complete means showing that you have included *all* the rules you need, that you haven't left any out. Showing this is beyond the scope of this book. The important point is that, happily, the proof system for SL is both sound and complete. This is not the case for all proof systems or all formal languages. Because it is true of SL, we can choose to give proofs or give truth tables—whichever is easier for the task at hand.

Now that we know that the truth table method is interchangeable with the method of derivations, you can chose which method you want to use for any given problem. Students often prefer to use truth tables, because they can be produced purely mechanically, and that seems 'easier'. However, we have already seen that truth tables become impossibly large after just a few sentence letters. On the other hand, there are a couple situations where using proofs simply isn't possible. We syntactically defined a contingent sentence as a sentence that couldn't be proven to be a tautology or a contradiction. There is no practical way to prove this kind of negative statement. We will never know if there isn't some proof out there that a statement is a contradiction and we just haven't found it yet. We have nothing to do in this situation but resort to truth tables. Similarly, we can use derivations to prove two sentences equivalent, but what if we want to prove that they are *not* equivalent? We have no way of proving that we will never find the relevant proof. So we have to fall back on truth tables again.

Table 20.2 summarizes when it is best to give proofs and when it is best to give truth tables.

Logical property	To prove it present	To prove it absent
Being a theorem	Derive the sentence	Find the false line in the truth table for the sentence
Being a contradiction	Derive the negation of the sentence	Find the true line in the truth table for the sentence
Contingency	Find a false line and a true line in the truth table for the sentence	Prove the sentence or its negation
Equivalence	Derive each sentence from the other	Find a line in the truth tables for the sentence where they have different values
Consistency	Find a line in truth table for the sentence where they all are true	Derive a contradiction from the sentences
Validity	Derive the conclusion from the premises	Find no line in the truth table where the premises are true and the conclusion false.

Table 20.2: When to provide a truth table and when to provide a proof.

Practice exercises

A. Use either a derivation or a truth table for each of the following.

1. Show that $A \rightarrow [((B \wedge C) \vee D) \rightarrow A]$ is a theorem.
2. Show that $A \rightarrow (A \rightarrow B)$ is not a theorem.
3. Show that the sentence $A \rightarrow \neg A$ is not a contradiction.
4. Show that the sentence $A \leftrightarrow \neg A$ is a contradiction.
5. Show that the sentence $\neg(W \rightarrow (J \vee J))$ is contingent.
6. Show that the sentence $\neg(X \vee (Y \vee Z)) \vee (X \vee (Y \vee Z))$ is not contingent.
7. Show that the sentence $B \rightarrow \neg S$ is equivalent to the sentence $\neg\neg B \rightarrow \neg S$.
8. Show that the sentence $\neg(X \vee O)$ is not equivalent to the sentence $X \wedge O$.
9. Show that the sentences $\neg(A \vee B)$, C, $C \rightarrow A$ are jointly inconsistent.
10. Show that the sentences $\neg(A \vee B)$, $\neg B$, $B \rightarrow A$ are jointly consistent.

11. Show that $\neg(A \vee (B \vee C))\ \therefore \neg C$ is valid.

12. Show that $\neg(A \wedge (B \vee C))\ \therefore \neg C$ is invalid.

B. Use either a derivation or a truth table for each of the following.

1. Show that $A \rightarrow (B \rightarrow A)$ is a theorem.

2. Show that $\neg(((N \leftrightarrow Q) \vee Q) \vee N)$ is not a theorem.

3. Show that $Z \vee (\neg Z \leftrightarrow Z)$ is contingent.

4. show that $(L \leftrightarrow ((N \rightarrow N) \rightarrow L)) \vee H$ is not contingent.

5. Show that $(A \leftrightarrow A) \wedge (B \wedge \neg B)$ is a contradiction.

6. Show that $(B \leftrightarrow (C \vee B))$ is not a contradiction.

7. Show that $((\neg X \leftrightarrow X) \vee X)$ is equivalent to X.

8. Show that $F \wedge (K \wedge R)$ is not equivalent to $(F \leftrightarrow (K \leftrightarrow R))$.

9. Show that the sentences $\neg(W \rightarrow W)$, $(W \leftrightarrow W) \wedge W$, $E \vee (W \rightarrow \neg(E \wedge W))$ are jointly inconsistent.

10. Show that the sentences $\neg R \vee C$, $(C \wedge R) \rightarrow \neg R$, $(\neg(R \vee R) \rightarrow R)$ are jointly consistent.

11. Show that $\neg\neg(C \leftrightarrow \neg C)$, $((G \vee C) \vee G)\ \therefore\ ((G \rightarrow C) \wedge G)$ is valid.

12. Show that $\neg\neg L$, $(C \rightarrow \neg L) \rightarrow C)\ \therefore\ \neg C$ is invalid.

PART V

Predicate logic

21 Building blocks of PL

21.1 The need to decompose sentences

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

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

To symbolize it in SL, we might offer a symbolization key:

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

And the argument itself becomes:

$$L, A \therefore F$$

This is *invalid* in SL, but the original English argument is clearly valid.

The problem is not that we have made a mistake while symbolizing the argument. This is the best symbolization we can give *in SL*. The problem lies with SL itself. 'All logicians wear funny hats' is about both logicians and hat-wearing. By not retaining this structure in our symbolization, we lose the connection between Willard's being a logician and Willard's wearing a hat.

The basic units of SL are sentence letters, and SL cannot decompose these. To symbolize arguments like the preceding one, we will have to develop a new logical language which will allow us to *split the atom*. We will call this language *predicate*, or *PL*.

The details of PL will be explained throughout this chapter, but here is the basic idea for splitting the atom.

First, we have *names*. In PL, we indicate these with lowercase italic letters. For instance, we might let 'b' stand for Bertie, or let 'i' stand for Willard.

Second, we have predicates. English predicates are expressions like '_____ is a dog' or '_____ is a logician'. These are not complete sentences by themselves. In order to make a complete sentence, we need to fill in the gap. We need to say something like 'Bertie is a dog' or 'Willard is a logician'. In PL, we indicate predicates with uppercase italic letters. For instance, we might let the PL predicate 'D' symbolize the English predicate '_____ is a dog'. Then the expression 'Db' will be a sentence in PL, which symbolizes the English sentence 'Bertie is a dog'. Equally, we might let the PL predicate 'L' symbolize the English predicate '_____ is a logician'. Then the expression 'Li' will symbolize the English sentence 'Willard is a logician'.

Third, we have quantifiers. For instance, '$\exists$' will roughly convey 'There is at least one ...'. So we might symbolize the English sentence 'there is a dog' with the PL sentence '$\exists x\, Dx$', which we would naturally read out-loud as 'there is at least one thing, x, such that x is a dog'.

That is the general idea, but PL is significantly more subtle than SL, so we will come at it slowly.

21.2 Names

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 'Bertie' is a singular term, because it refers to a specific terrier. Likewise, the phrase 'Philip's dog Bertie' is a singular term, because it refers to a specific little terrier.

Proper names are a particularly important kind of singular term. These are expressions that pick out individuals without describing them. 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 'Hilary' is used as a singular term, you might guess that it refers to a woman. You might, though, be guessing wrongly. Indeed, the name does not necessarily mean that the person referred to is even a person: Hilary might be a giraffe, for all you could tell just from the name.

In PL, our NAMES are lower-case letters 'a' through to 'r'. We can add subscripts if we want to use some letter more than once. So here are some singular terms in PL:

$$a, b, c, \ldots, r, a_1, f_{32}, j_{390}, m_{12}$$

These should be thought of along the lines of proper names in English, but with one difference. 'Tim Button' is a proper name, but there are several people with this name. (Equally, there are at least two people with the name 'P.D. Magnus'.) We live with this kind of ambiguity in English, allowing context to individuate the fact that 'Tim Button' refers to an author of this book, and not some other Tim. In PL, we do not tolerate any such ambiguity. Each name must pick out *exactly* one thing. (However, two different names may pick out the same thing.)

As with SL, we can provide symbolization keys. These indicate, temporarily, what a name will pick out. So we might offer:

e:	Elsa
g:	Gregor
m:	Marybeth

21.3 Predicates

The simplest predicates are properties of individuals. They are things you can say about an object. Here are some examples of English predicates:

> _____ is a dog
> _____ is a member of Monty Python
> A piano fell on _____

In general, you can think about predicates as things which combine with singular terms to make sentences. 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 obtain any of three different predicates:

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

In PL, PREDICATES are capital letters A through Z, with or without subscripts. We might write a symbolization key for predicates thus:

Ax: $\underline{\qquad}_x$ is angry
Hx: $\underline{\qquad}_x$ is happy

(Why the subscripts on the gaps? We will return to this in §23.)

If we combine our two symbolization keys, we can start to symbolize some English sentences that use these names and predicates in combination. For example, consider the English sentences:

1. Elsa is angry.
2. Gregor and Marybeth are angry.
3. If Elsa is angry, then so are Gregor and Marybeth.

Sentence 1 is straightforward: we symbolize it by 'Ae'.

Sentence 2: this is a conjunction of two simpler sentences. The simple sentences can be symbolized just by 'Ag' and 'Am'. Then we help ourselves to our resources from SL, and symbolize the entire sentence by '$Ag \wedge Am$'. This illustrates an important point: PL has all of the truth-functional connectives of SL.

Sentence 3: this is a conditional, whose antecedent is sentence 1 and whose consequent is sentence 2, so we can symbolize this with '$Ae \rightarrow (Ag \wedge Am)$'.

21.4 Quantifiers

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

4. Everyone is happy.

5. Someone is angry.

It might be tempting to symbolize sentence 4 as '$He \land Hg \land Hm$'. Yet this would only say that Elsa, Gregor, and Marybeth are happy. We want to say that *everyone* is happy, even those with no names. 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. In PL, variables are italic lowercase letters 'w' through 'z', with or without subscripts. So we might symbolize sentence 4 as '$\forall x\, Hx$'. The variable 'x' is serving as a kind of placeholder. The expression '$\forall x$' intuitively means that you can pick anyone and put them in as 'x'. The subsequent 'Hx' indicates, of that thing you picked out, that it is happy.

It should be pointed out that there is no special reason to use 'x' rather than some other variable. The sentences '$\forall x\, Hx$', '$\forall y\, Hy$', '$\forall z\, Hz$', and '$\forall x_5 Hx_5$' use different variables, but they will all be logically equivalent.

To symbolize sentence 5, we introduce another new symbol: the EXISTENTIAL QUANTIFIER, '$\exists$'. Like the universal quantifier, the existential quantifier requires a variable. Sentence 5 can be symbolized by '$\exists x\, Ax$'. Whereas '$\forall x\, Ax$' is read naturally as 'for all x, x is angry', '$\exists x\, Ax$' is read naturally as 'there is something, x, such that x is angry'. Once again, the variable is a kind of placeholder; we could just as easily have symbolized sentence 5 by '$\exists z\, Az$', '$\exists w_{256}\, Aw_{256}$', or whatever.

Some more examples will help. Consider these further sentences:

6. No one is angry.
7. There is someone who is not happy.
8. Not everyone is happy.

Sentence 6 can be paraphrased as, 'It is not the case that someone is angry'. We can then symbolize it using negation and an existential quantifier: '$\neg\exists x\, Ax$'. Yet sentence 6 could also be paraphrased as, 'Everyone is not angry'. With this in mind, it can be symbolized using negation and a universal quantifier: '$\forall x\, \neg Ax$'. Both of these are acceptable symbolizations. Indeed, it will transpire that, in general, $\forall x\, \neg\mathscr{A}$ is logically equivalent to $\neg\exists x\, \mathscr{A}$. (Notice that we have here returned to the practice of using '$\mathscr{A}$' as a metavariable, from §7.) Symbolizing a sentence one way, rather than the other, might seem more 'natural' in some contexts, but it is not much more than a matter of taste.

Sentence 7 is most naturally paraphrased as, 'There is some x, such that x is not happy'. This then becomes '$\exists x\, \neg Hx$'. Of course, we could equally have written '$\neg\forall x\, Hx$', which we would naturally read as 'it is not the case that everyone is happy'. Whenever someone is not happy, it won't be true that *everyone* is happy; and, whenever it's not true that everyone is happy, there will be someone who is not happy. So '$\exists x\, \neg Hx$' and '$\neg\forall x\, Hx$' say the same thing. So 7 and **??** say the same thing. So either '$\exists x\, \neg Hx$' or '$\neg\forall x\, Hx$' would be a perfectly adequate symbolization of sentence 8.

21.5 Domains

Given the symbolization key we have been using, '$\forall x\, Hx$' symbolizes '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 usually mean something more modest: everyone now in the building, everyone enrolled in the ballet class, or whatever.

In order to eliminate this ambiguity, we will need to specify a DOMAIN. The domain is the collection of things that we are talking about. So if we want to talk about people in Chicago, we define the domain to be people in Chicago. We write this at the beginning of the symbolization key, like this:

domain: people in Chicago

The quantifiers *range over* the domain. Given this domain, '$\forall x$' is to be read roughly as 'Every person in Chicago is such that...' and '$\exists x$' is to be read roughly as 'Some person in Chicago is such that....'

In PL, the domain must always include at least one thing. Moreover, in English we can infer 'something is angry' from 'Gregor is angry'. In PL, then, we will want to be able to infer '$\exists x\, Ax$' from 'Ag'. So we will insist that each name must pick out exactly one thing in the domain. If we want to name people in places beside Chicago, then we need to include those people in the domain.

A domain must have *at least* one member. A name must pick out *exactly* one member of the domain, but a member of the domain may be picked out by one name, many names, or none at all.

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

domain: the Eiffel Tower
Px: $\_\_\_\_\__x$ is in Paris.

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

Non-referring terms

In PL, each name must pick out exactly one member of the domain. A name cannot refer to more than one thing—it is a *singular* term. Each name 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:

9. Chimera is angry.
10. Chimera is not angry.

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

domain:	creatures on Earth
Ax:	$\_\_\_\_\__{x}$ is angry.
c:	chimera

We could then symbolize sentence 9 as Ac and sentence 10 as $\neg Ac$.

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

One option is to say that sentence 9 is not true, because there is no chimera. If sentence 9 is false because it talks about a non-existent thing, then sentence 10 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 9 is *meaningless* because it talks about a non-existent thing. So Ac would be a meaningful expression in PL 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. **??**.) The important point for now is that each name of PL *must* refer to something in the domain, although the domain can contain any things we like. If we want to symbolize arguments about mythological creatures, then we must define a domain that includes them. This option is important if we want to consider the logic of stories. We can symbolize a sentence like 'Sherlock Holmes lived at 221B Baker Street' by including fictional characters like Sherlock Holmes in our domain.

22 Sentences with one quantifier

We now have all of the pieces of PL. Symbolizing more complicated sentences will only be a matter of knowing the right way to combine predicates, names, quantifiers, and connectives. There is a knack to this, and there is no substitute for practice.

22.1 Common quantifier phrases

Consider these sentences:

1. Every coin in my pocket is a quarter.
2. Some coin on the table is a dime.
3. Not all the coins on the table are dimes.
4. None of the coins in my pocket are dimes.

In providing a symbolization key, we need to specify a domain. Since we are talking about coins in my pocket and on the table, the domain must at least contain all of those coins. Since we are not talking about anything besides coins, we let the domain be all coins. Since we are not talking about any specific coins, we do not need to deal with any names. So here is our key:

domain:	all coins
Px:	$\underline{\quad\quad}_x$ is in my pocket
Tx:	$\underline{\quad\quad}_x$ is on the table
Qx:	$\underline{\quad\quad}_x$ is a quarter
Dx:	$\underline{\quad\quad}_x$ is a dime

Sentence 1 is most naturally symbolized using a universal quantifier. The universal quantifier says something about everything in the domain, not just about the coins in my pocket. Sentence 1 can be paraphrased as 'for any coin, *if* that coin is in my pocket *then* it is a quarter'. So we can symbolize it as '$\forall x(Px \rightarrow Qx)$'.

Since sentence 1 is about coins that are both in my pocket *and* that are quarters, it might be tempting to symbolize it using a conjunction. However, the sentence '$\forall x(Px \wedge Qx)$' would symbolize the sentence 'every coin is both a quarter and in my pocket'. This obviously means something very different than sentence 1. And so we see:

A sentence can be symbolized as $\forall x(\mathscr{F}x \rightarrow \mathscr{G}x)$ if it can be paraphrased in English as 'every F is G'.

Sentence 2 is most naturally symbolized using an existential quantifier. It can be paraphrased as 'there is some coin which is both on the table and which is a dime'. So we can symbolize 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. Suppose we had instead written '$\exists x(Tx \rightarrow Dx)$'. That would mean that there is some object in the domain of which '$(Tx \rightarrow Dx)$' is true. Recall that, in SL, $\mathscr{A} \rightarrow \mathscr{B}$ is logically equivalent (in SL) to $\neg\mathscr{A} \vee \mathscr{B}$. This equivalence will also hold in PL. So '$\exists x(Tx \rightarrow Dx)$' is true if there is some object in the domain, such that '$(\neg Tx \vee Dx)$' is true of that object. That is, '$\exists x(Tx \rightarrow Dx)$' 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 lots of other places. So it is *very easy* for '$\exists x(Tx \rightarrow Dx)$' to be 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 tends to say something very weak indeed. As a general rule of thumb, do not put conditionals in the scope of existential quantifiers unless you are sure that you need one.

A sentence can be symbolized as $\exists x(\mathscr{F}x \wedge \mathscr{G}x)$ if it can be paraphrased in English as 'some F is G'.

Sentence 3 can be paraphrased as, 'It is not the case that every coin on the table is a dime'. So we can symbolize it by '$\neg\forall x(Tx \rightarrow Dx)$'. You might look at sentence 3 and paraphrase it instead as, 'Some coin on the table is not a dime'. You would then symbolize it by '$\exists x(Tx \wedge \neg Dx)$'. Although it is probably not immediately obvious yet, these two sentences are logically equivalent. (This is due to the logical equivalence between $\neg\forall x\,\mathscr{A}$ and $\exists x\neg\mathscr{A}$, mentioned in §21, along with the equivalence between $\neg(\mathscr{A} \rightarrow \mathscr{B})$ and $\mathscr{A} \wedge \neg\mathscr{B}$.)

Sentence 4 can be paraphrased as, 'It is not the case that there is some dime in my pocket'. This can be symbolized by '$\neg\exists x(Px \wedge Dx)$'. It might also be paraphrased as, 'Everything in my pocket is a non-dime', and then could be symbolized by '$\forall x(Px \rightarrow \neg Dx)$'. Again the two symbolizations are logically equivalent; both are correct symbolizations of sentence 4.

22.2 Empty predicates

In §21, we emphasized that a name must pick out exactly one object in the domain. However, a predicate need not apply to anything in the domain. A predicate that applies to nothing in the domain is called an EMPTY PREDICATE. This is worth exploring.

Suppose we want to symbolize these two sentences:

5. Every monkey knows sign language
6. Some monkey knows sign language

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

domain: animals
Mx: $\_\_\_\_\__x$ is a monkey.
Sx: $\_\_\_\_\__x$ knows sign language.

Sentence 5 can now be symbolized by '$\forall x(Mx \rightarrow Sx)$'. Sentence 6 can be symbolized as '$\exists x(Mx \wedge Sx)$'.

It is tempting to say that sentence 5 *entails* sentence 6. That is, we might think that it is impossible for it to be the case that every monkey knows sign language, without its also being the case that some monkey knows sign language, but this would be a mistake. It is possible for the sentence '$\forall x(Mx \rightarrow Sx)$' 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*. If there were no monkeys at all (in the domain), then '$\forall x(Mx \rightarrow Sx)$' would be *vacuously* true: take any monkey you like—it knows sign language! But if there were no monkeys at all (in the domain), then '$\exists x(Mx \wedge Sx)$' would be false.

Another example will help to bring this home. Suppose we extend the above symbolization key, by adding:

Rx: $\_\_\_\_\__x$ is a refrigerator

Now consider the sentence '$\forall x(Rx \rightarrow Mx)$'. This symbolizes 'every refrigerator is a monkey'. This sentence is true, given our symbolization key, which is counterintuitive, since we (presumably) do not want to say that there are a whole bunch of refrigerator monkeys. It is important to remember, though, that '$\forall x(Rx \rightarrow Mx)$' is true iff any member of the domain that is a refrigerator is a monkey. Since the domain is *animals*, there are no refrigerators in the domain. Again, then, the sentence is *vacuously* true.

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

When $\mathscr{F}$ is an empty predicate, a sentence $\forall x(\mathscr{F}x \rightarrow \ldots)$ will be vacuously true.

22.3 Picking a domain

The appropriate symbolization of an English language sentence in PL will depend on the symbolization key. Choosing a key can be difficult. Suppose we want to symbolize the English sentence:

7. Every rose has a thorn.

We might offer this symbolization key:

Rx:	$\_\_\_\_\_\__x$ is a rose
Tx:	$\_\_\_\_\_\__x$ has a thorn

It is tempting to say that sentence 7 should be symbolized as '$\forall x(Rx \rightarrow Tx)$', but we have not yet chosen a domain. If the domain contains all roses, this would be a good symbolization. Yet if the domain is merely *things on my kitchen table*, then '$\forall x(Rx \rightarrow Tx)$' would only come close to covering the fact 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. This is not what we want. To symbolize sentence 7 adequately, we need to include all the roses in the domain, but now we have two options.

First, we can restrict the domain to include all roses but *only* roses. Then sentence 7 can, if we like, be symbolized with '$\forall x\, Tx$'. This is true iff everything in the domain has a thorn; since the domain is just the roses, this is true iff every rose has a thorn. By restricting the domain, we have been able to symbolize our English sentence with a very short sentence of PL. So this approach can save us trouble, if every sentence that we want to deal with is about roses.

Second, we can let the domain contain things besides roses: rhododendrons; rats; rifles; whatevers, and we will certainly need to include a more expansive domain if we simultaneously want to symbolize sentences like:

8. Every cowboy sings a sad, sad song.

Our domain must now include both all the roses (so that we can symbolize sentence 7) and all the cowboys (so that we can symbolize sentence 8). So we might offer the following symbolization key:

domain:	people and plants
Cx:	$\_\_\_\_\_\__x$ is a cowboy
Sx:	$\_\_\_\_\_\__x$ sings a sad, sad song
Rx:	$\_\_\_\_\_\__x$ is a rose
Tx:	$\_\_\_\_\_\__x$ has a thorn

Now we will have to symbolize sentence 7 with '$\forall x(Rx \rightarrow Tx)$', since '$\forall x\, Tx$' would symbolize the sentence 'every person or plant has a thorn'. Similarly, we will have to symbolize sentence 8 with '$\forall x(Cx \rightarrow Sx)$'.

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

22.4 The utility of paraphrase

When symbolizing English sentences in PL, it is important to understand the structure of the sentences you want to symbolize. What matters is the final symbolization in PL, and sometimes you will be able to move from an English language sentence directly to a sentence of PL. 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 easily symbolize directly in PL.

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

domain:	people
Bx:	$\_\_\_\_\__x$ is a bassist.
Rx:	$\_\_\_\_\__x$ is a rock star.
k:	Kim Deal

Now consider these sentences:

9. If Kim Deal is a bassist, then she is a rock star.
10. If a person is a bassist, then she is a rock star.

The same words appear as the consequent in sentences 9 and 10 ('. . . she is a rock star'), but they mean very different things. To make this clear, it often helps to paraphrase the original sentences, removing pronouns.

Sentence 9 can be paraphrased as, 'If Kim Deal is a bassist, then *Kim Deal* is a rockstar'. This can obviously be symbolized as '$Bk \rightarrow Rk$'.

Sentence 10 must be paraphrased differently: 'If a person is a bassist, then *that person* is a rock star'. This sentence is not about any particular person, so we need a variable. As an intermediate step, we can paraphrase this as, 'For any person x, if x is a bassist, then x is a rockstar'. Now this can be symbolized as '$\forall x(Bx \rightarrow Rx)$'. This is the same sentence we would have used to symbolize 'Everyone who is a bassist is a rock star'. On reflection, that is surely true iff sentence 10 is true, as we would hope.

Consider these further sentences:

11. If anyone is a bassist, then Kim Deal is a rock star.
12. If anyone is a bassist, then she is a rock star.

The same words appear as the antecedent in sentences 11 and 12 ('If anyone is a bassist. . .'), but it can be tricky to work out how to symbolize these two uses. Again, paraphrase will come to our aid.

Sentence 11 can be paraphrased, 'If there is at least one bassist, then Kim Deal is a rock star'. It is now clear that this is a conditional whose antecedent is a quantified expression; so we can symbolize the entire sentence with a conditional as the main logical operator: '$\exists xBx \rightarrow Rk$'.

Sentence 12 can be paraphrased, 'For all people x, if x is a bassist, then x is a rock star'. Or, in more natural English, it can be paraphrased by 'All bassists are rock stars'. It is best symbolized as '$\forall x(Bx \rightarrow Rx)$', just like sentence 10.

The moral is that the English words 'any' and 'anyone' should typically be symbolized using quantifiers, and if you are having a hard time determining whether to use an existential or a universal quantifier, try paraphrasing the sentence with an English sentence that uses words *besides* 'any' or 'anyone'.

22.5 Quantifiers and scope

Continuing the example, suppose we want to symbolize these sentences:

13. If everyone is a bassist, then Lars is a bassist
14. Everyone is such that, if they are a bassist, then Lars is a bassist.

To symbolize these sentences, we will have to add a new name to the symbolization key, namely:

> l: Lars

Sentence 13 is a conditional, whose antecedent is 'everyone is a bassist', so we will symbolize it with '$\forall x\, Bx \to Bl$'. This sentence is *necessarily* true: if *everyone* is indeed a bassist, then take any one you like—for example Lars—and he will be a bassist.

Sentence 14, by contrast, might best be paraphrased by 'every person x is such that, if x is a bassist, then Lars is a bassist'. This is symbolized by '$\forall x(Bx \to Bl)$'. This sentence is false; Kim Deal is a bassist. So 'Bk' is true, but Lars is not a bassist, so 'Bl' is false. Accordingly, '$Bk \to Bl$' will be false, so '$\forall x(Bx \to Bl)$' will be false as well.

In short, '$\forall x Bx \to Bl$' and '$\forall x(Bx \to Bl)$' are very different sentences. We can explain the difference in terms of the *scope* of the quantifier. The scope of quantification is very much like the scope of negation, which we considered when discussing SL, and it will help to explain it in this way.

In the sentence '$\neg Bk \to Bl$', the scope of '$\neg$' is just the antecedent of the conditional. We are saying something like: if 'Bk' is false, then 'Bl' is true. Similarly, in the sentence '$\forall x Bx \to Bl$', the scope of '$\forall x$' is just the antecedent of the conditional. We are saying something like: if 'Bx' is true of *everything*, then 'Bl' is also true.

In the sentence '$\neg(Bk \to Bl)$', the scope of '$\neg$' is the entire sentence. We are saying something like: '$(Bk \to Bl)$' is false. Similarly, in the sentence '$\forall x(Bx \to Bl)$', the scope of '$\forall x$' is the entire sentence. We are saying something like: '$(Bx \to Bl)$' is true of *everything*.

The moral of the story is simple. When you are using conditionals, be very careful to make sure that you have sorted out the scope correctly.

Ambiguous predicates

Suppose we just want to symbolize this sentence:

15. Adina is a skilled surgeon.

Let the domain be people, let Kx mean 'x is a skilled surgeon', and let a mean Adina. Sentence 15 is simply Ka.

Suppose instead that we want to symbolize 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:

domain:	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 symbolized 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 symbolize 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 name (let c mean Carol). Then the argument becomes:

$$(Rc \wedge Kc) \wedge Tc$$
$$\therefore\ Tc \wedge Kc$$

This symbolization is a disaster! It takes what in English is a terrible argument and symbolizes it as a valid argument in PL. The problem is that there is a difference between being *skilled as a surgeon* and *skilled as a tennis player*. Symbolizing 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:

$$(Rc \wedge K_1c) \wedge Tc$$
$$\therefore\ Tc \wedge K_2c$$

Like the English language argument it symbolizes, 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 15 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 symbolizing an argument that is just about dogs, it is fine to define a predicate that means '*x* is big.' If the domain includes dogs and mice, however, it is probably best to make the predicate mean '*x* is big for a dog.'

Practice exercises

A. Here are the syllogistic figures identified by Aristotle and his successors, along with their medieval names:

- **Barbara.** All G are F. All H are G. So: All H are F
- **Celarent.** No G are F. All H are G. So: No H are F
- **Ferio.** No G are F. Some H is G. So: Some H is not F
- **Darii.** All G are F. Some H is G. So: Some H is F.
- **Camestres.** All F are G. No H are G. So: No H are F.
- **Cesare.** No F are G. All H are G. So: No H are F.
- **Baroko.** All F are G. Some H is not G. So: Some H is not F.
- **Festino.** No F are G. Some H are G. So: Some H is not F.
- **Datisi.** All G are F. Some G is H. So: Some H is F.
- **Disamis.** Some G is F. All G are H. So: Some H is F.
- **Ferison.** No G are F. Some G is H. So: Some H is not F.
- **Bokardo.** Some G is not F. All G are H. So: Some H is not F.
- **Camenes.** All F are G. No G are H So: No H is F.
- **Dimaris.** Some F is G. All G are H. So: Some H is F.
- **Fresison.** No F are G. Some G is H. So: Some H is not F.

Symbolize each argument in PL.

B. Using the following symbolization key:

domain:	people
Kx:	$\rule{1cm}{0.4pt}_x$ knows the combination to the safe
Sx:	$\rule{1cm}{0.4pt}_x$ is a spy
Vx:	$\rule{1cm}{0.4pt}_x$ is a vegetarian
h:	Hofthor
i:	Ingmar

symbolize the following sentences in PL:

1. Neither Hofthor nor Ingmar is a vegetarian.
2. No spy knows the combination to the safe.
3. No one knows the combination to the safe unless Ingmar does.
4. Hofthor is a spy, but no vegetarian is a spy.

C. Using this symbolization key:

domain:	all animals
Ax:	$\underline{\qquad}_{x}$ is an alligator.
Mx:	$\underline{\qquad}_{x}$ is a monkey.
Rx:	$\underline{\qquad}_{x}$ is a reptile.
Zx:	$\underline{\qquad}_{x}$ lives at the zoo.
a:	Amos
b:	Bouncer
c:	Cleo

symbolize each of the following sentences in PL:

1. Amos, Bouncer, and Cleo all live at the zoo.
2. Bouncer is a reptile, but not an alligator.
3. Some reptile lives at the zoo.
4. Every alligator is a reptile.
5. Any animal that lives at the zoo is either a monkey or an alligator.
6. There are reptiles which are not alligators.
7. If any animal is an reptile, then Amos is.
8. If any animal is an alligator, then it is a reptile.

D. For each argument, write a symbolization key and symbolize the argument in PL.

1. Willard is a logician. All logicians wear funny hats. So Willard wears a funny hat
2. 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.
3. All my dreams are black and white. Old TV shows are in black and white. Therefore, some of my dreams are old TV shows.
4. 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.
5. 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.
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.

23 Multiple generality

So far, we have only considered sentences that require one-place predicates and one quantifier. The full power of PL really comes out when we start to use many-place predicates and multiple quantifiers.

23.1 Many-placed predicates

All of the predicates that we have considered so far concern properties that objects might have. Those predicates have one gap in them, and to make a sentence, we simply need to slot in one term. They are ONE-PLACE predicates.

However, other predicates concern the *relation* between two things. Here are some examples of relational predicates in English:

_____ loves _____
_____ is to the left of _____
_____ is in debt to _____

These are TWO-PLACE predicates. They need to be filled in with two terms in order to make a sentence. Conversely, if we start with an English sentence containing many singular terms, we can remove two singular terms, to obtain different two-place predicates. Consider the sentence 'Vinnie borrowed the family car from Nunzio'. By deleting two singular terms, we can obtain any of three different two-place predicates

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

and by removing all three singular terms, we obtain a THREE-PLACE predicate:

_____ borrowed _____ from _____

Indeed, there is no in principle upper limit on the number of places that our predicates may contain.

Now there is a little foible with the above. We have used the same symbol, '_____', to indicate a gap formed by deleting a term from a sentence. However (as Frege emphasized), these are *different* gaps. To obtain a sentence, we can fill them in with the same term, but we can equally fill them in with different terms, and in various different orders. The following are all perfectly good sentences, and they all mean very different things:

Karl loves Karl
Karl loves Imre
Imre loves Karl
Imre loves Imre

The point is that we need to keep track of the gaps in predicates, so that we can keep track of how we are filling them in.

To keep track of the gaps, we will label them. The labelling conventions we will adopt are best explained by example. Suppose we want to symbolize the following sentences:

1. Karl loves Imre.
2. Imre loves himself.
3. Karl loves Imre, but not vice versa.
4. Karl is loved by Imre.

We will start with the following representation key:

domain:	people
i:	Imre
k:	Karl
Lxy:	$\_\_\_\_\__x$ loves $\_\_\_\_\__y$

Sentence 1 will now be symbolized by 'Lki'.

Sentence 2 can be paraphrased as 'Imre loves Imre'. It can now be symbolized by 'Lii'.

Sentence 3 is a conjunction. We might paraphrase it as 'Karl loves Imre, and Imre does not love Karl'. It can now be symbolized by '$Lki \land \neg Lik$'.

Sentence 4 might be paraphrased by 'Imre loves Karl'. It can then be symbolized by 'Lik'. Of course, this slurs over the difference in tone between the active and passive voice; such nuances are lost in PL.

This last example, though, highlights something important. Suppose we add to our symbolization key the following:

Mxy: $\_\_\_\_\__y$ loves $\_\_\_\_\__x$

Here, we have used the same English word ('loves') as we used in our symbolization key for 'Lxy'. However, we have swapped the order of the *gaps* around (look closely at those little subscripts!) So 'Mki' and 'Lik' now *both* symbolize 'Imre loves Karl'. 'Mik' and 'Lki' now *both* symbolize 'Karl loves Imre'. Since love can be unrequited, these are very different claims.

The moral is simple. When we are dealing with predicates with more than one place, we need to pay careful attention to the order of the places.

23.2 The order of quantifiers

Consider the sentence 'everyone loves someone'. This is potentially ambiguous. It might mean either of the following:

5. For every person x, there is some person that x loves
6. There is some particular person whom every person loves

Sentence 5 can be symbolized by '$\forall x \exists y\, Lxy$', and would be true of a love-triangle. For example, suppose that our domain of discourse is restricted to Imre, Juan and Karl. Suppose also that Karl loves Imre but not Juan, that Imre loves Juan but not Karl, and that Juan loves Karl but not Imre. Then sentence 5 is true.

Sentence 6 is symbolized by '$\exists y \forall x\, Lxy$'. Sentence 6 is *not* true in the situation just described. Again, suppose that our domain of discourse is restricted to Imre, Juan and Karl. This requires that all of Juan, Imre and Karl converge on (at least) one object of love.

The point of the example is to illustrate that the order of the quantifiers matters a great deal. Indeed, to switch them around is called a *quantifier shift fallacy*. Here is an example, which comes up in various forms throughout the philosophical literature:

	For every person, there is some truth they cannot know.	($\forall\exists$)
∴	There is some truth that no person can know.	($\exists\forall$)

This argument form is obviously invalid. It's just as bad as:[1]

	Every dog has its day.	($\forall\exists$)
∴	There is a day for all the dogs.	($\exists\forall$)

The order of quantifiers is also important in definitions in mathematics. For instance, there is a big difference between pointwise and uniform continuity of functions:

- ▹ A function f is *pointwise continuous* if

$$\forall\epsilon\forall x\forall y\exists\delta(|x-y|<\delta\rightarrow|f(x)-f(y)|<\epsilon)$$

- ▹ A function f is *uniformly continuous* if

$$\forall\epsilon\exists\delta\forall x\forall y(|x-y|<\delta\rightarrow|f(x)-f(y)|<\epsilon)$$

The moral is: take great care with the order of quantification.

[1] Thanks to Rob Trueman for the example.

23.3 Stepping-stones to symbolization

Once we have the possibility of multiple quantifiers and many-place predicates, representation in PL can quickly start to become a bit tricky. When you are trying to symbolize a complex sentence, we recommend laying down several stepping stones. As usual, this idea is best illustrated by example. Consider this representation key:

domain:	people and dogs
Dx:	$\_\_\_\_\__x$ is a dog
Fxy:	$\_\_\_\_\__x$ is a friend of $\_\_\_\_\__y$
Oxy:	$\_\_\_\_\__x$ owns $\_\_\_\_\__y$
g:	Geraldo

Now let's try to symbolize these sentences:

7. Geraldo is a dog owner.
8. Someone is a dog owner.
9. All of Geraldo's friends are dog owners.
10. Every dog owner is a friend of a dog owner.
11. Every dog owner's friend owns a dog of a friend.

Sentence 7 can be paraphrased as, 'There is a dog that Geraldo owns'. This can be symbolized by '$\exists x(Dx \wedge Ogx)$'.

Sentence 8 can be paraphrased as, 'There is some y such that y is a dog owner'. Dealing with part of this, we might write '$\exists y(y$ is a dog owner$)$'. Now the fragment we have left as 'y is a dog owner' is much like sentence 7, except that it is not specifically about Geraldo. So we can symbolize sentence 8 by:

$$\exists y \exists x(Dx \wedge Oyx)$$

We should pause to clarify something here. In working out how to symbolize the last sentence, we wrote down '$\exists y(y$ is a dog owner$)$'. To be very clear: this is *neither* an PL sentence *nor* an English sentence: it uses bits of PL ('$\exists$', 'y') and bits of English ('dog owner'). It is really is *just a stepping-stone* on the way to symbolizing the entire English sentence with a PL sentence. You should regard it as a bit of rough-working-out, on a par with the doodles that you might absent-mindedly draw in the margin of this book, whilst you are concentrating fiercely on some problem.

Sentence 9 can be paraphrased as, 'Everyone who is a friend of Geraldo is a dog owner'. Using our stepping-stone tactic, we might write

$$\forall x\big[Fxg \rightarrow x \text{ is a dog owner}\big]$$

Now the fragment that we have left to deal with, 'x is a dog owner', is structurally just like sentence 7. However, it would be a mistake for us simply to write

$$\forall x\big[Fxg \rightarrow \exists x(Dx \wedge Oxx)\big]$$

for we would here have a *clash of variables*. The scope of the universal quantifier, ‘$\forall x$’, is the entire conditional, so the ‘x’ in ‘Dx’ should be governed by that, but ‘Dx’ also falls under the scope of the existential quantifier ‘$\exists x$’, so the ‘x’ in ‘Dx’ should be governed by that. Now confusion reigns: which ‘x’ are we talking about? Suddenly the sentence becomes ambiguous (if it is even meaningful at all), and logicians hate ambiguity. The broad moral is that a single variable cannot serve two quantifier-masters simultaneously.

To continue our symbolization, then, we must choose some different variable for our existential quantifier. What we want is something like:

$$\forall x\big[Fxg \rightarrow \exists z(Dz \land Oxz)\big]$$

This adequately symbolizes sentence 9.

Sentence 10 can be paraphrased as ‘For any x that is a dog owner, there is a dog owner who x is a friend of’. Using our stepping-stone tactic, this becomes

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

Completing the symbolization, we end up with

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

Note that we have used the same letter, ‘z’, in both the antecedent and the consequent of the conditional, but that these are governed by two different quantifiers. This is ok: there is no clash here, because it is clear which quantifier that variable falls under. We might graphically represent the scope of the quantifiers thus:

$$\forall x\overbrace{\big[\overbrace{\exists z(Dz \land Oxz)}^{\text{scope of 1st ‘}\exists z\text{’}} \rightarrow \overbrace{\exists y(\overbrace{\exists z(Dz \land Oyz)}^{\text{scope of 2nd ‘}\exists z\text{’}} \land Fxy)\big]}^{\text{scope of ‘}\exists y\text{’}}}^{\text{scope of ‘}\forall x\text{’}}$$

This shows that no variable is being forced to serve two masters simultaneously.

Sentence 11 is the trickiest yet. First we paraphrase it as ‘For any x that is a friend of a dog owner, x owns a dog which is also owned by a friend of x’. Using our stepping-stone tactic, this becomes:

$$\forall x\big[x \text{ is a friend of a dog owner} \rightarrow \\ x \text{ owns a dog which is owned by a friend of } x\big]$$

Breaking this down a bit more:

$$\forall x\big[\exists y(Fxy \land y \text{ is a dog owner}) \rightarrow \\ \exists y(Dy \land Oxy \land y \text{ is owned by a friend of } x)\big]$$

And a bit more:

$$\forall x\big[\exists y(Fxy \wedge \exists z(Dz \wedge Oyz)) \rightarrow \exists y(Dy \wedge Oxy \wedge \exists z(Fzx \wedge Ozy))\big]$$

And we are done!

23.4 Supressed quantifiers

Logic can often help to get clear on the meanings of English claims, especially where the quantifiers are left implicit or their order is ambiguous or unclear. The clarity of expression and thinking afforded by PL can give you a significant advantage in argument, as can be seen in the following takedown by British political philosopher Mary Astell (1666–1731) of her contemporary, the theologian William Nicholls. In Discourse IV: The Duty of Wives to their Husbands of his *The Duty of Inferiors towards their Superiors, in Five Practical Discourses* (London 1701), Nicholls argued that women are naturally inferior to men. In the preface to the 3rd edition of her treatise *Some Reflections upon Marriage, Occasion'd by the Duke and Duchess of Mazarine's Case; which is also considered,* Astell responded as follows:

> 'Tis true, thro' Want of Learning, and of that Superior Genius which Men as Men lay claim to, she [Astell] was ignorant of the *Natural Inferiority* of our Sex, which our Masters lay down as a Self-Evident and Fundamental Truth. She saw nothing in the Reason of Things, to make this either a Principle or a Conclusion, but much to the contrary; it being Sedition at least, if not Treason to assert it in this Reign.
>
> For if by the Natural Superiority of their Sex, they mean that *every* Man is by Nature superior to *every* Woman, which is the obvious meaning, and that which must be stuck to if they would speak Sense, it wou'd be a Sin in *any* Woman to have Dominion over *any* Man, and the greatest Queen ought not to command but to obey her Footman, because no Municipal Laws can supersede or change the Law of Nature; so that if the Dominion of the Men be such, the *Salique Law,*[2] as unjust as *English Men* have ever thought it, ought to take place over all the Earth, and the most glorious Reigns in the *English, Danish, Castilian*, and other Annals, were wicked Violations of the Law of Nature!
>
> If they mean that *some* Men are superior to *some* Women this is no great Discovery; had they turn'd the Tables they might have seen that *some* Women are Superior to *some* Men. Or had they been pleased to remember their Oaths of Allegiance and Supremacy, they might have known that *One* Woman is superior to *All* the Men in these Nations, or

[2] The Salique law was the common law of France which prohibited the crown be passed on to female heirs.

> else they have sworn to very little purpose.[3] And it must not be suppos'd, that their Reason and Religion wou'd suffer them to take Oaths, contrary to the Laws of Nature and Reason of things.[4]

We can symbolize the different interpretations Astell offers of Nicholls' claim that men are superior to women: He either meant that every man is superior to every woman, i.e.,

$$\forall x(Mx \rightarrow \forall y(Wy \rightarrow Sxy))$$

or that some men are superior to some women,

$$\exists x(Mx \wedge \exists y(Wy \wedge Sxy)).$$

The latter is true, but so is

$$\exists y(Wy \wedge \exists x(Mx \wedge Syx)).$$

(some women are superior to some men), so that would be "no great discovery." In fact, since the Queen is superior to all her subjects, it's even true that some woman is superior to every man, i.e.,

$$\exists y(Wy \wedge \forall x(Mx \rightarrow Syx)).$$

But this is incompatible with the "obvious meaning" of Nicholls' claim, i.e., the first reading. So what Nicholls claims amounts to treason against the Queen!

Practice exercises

A. Using this symbolization key:

domain:	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

symbolize each of the following sentences in PL:

1. If Cleo loves Bouncer, then Bouncer is a monkey.
2. If both Bouncer and Cleo are alligators, then Amos loves them both.

[3] In 1706, England was ruled by Queen Anne.

[4] Mary Astell, *Reflections upon Marriage*, 1706 Preface, iii–iv, and Mary Astell, *Political Writings*, ed. Patricia Springborg, Cambridge University Press, 1996, 9–10.

3. Cleo loves a reptile.
4. Bouncer loves all the monkeys that live at the zoo.
5. All the monkeys that Amos loves love him back.
6. Every monkey that Cleo loves is also loved by Amos.
7. There is a monkey that loves Bouncer, but sadly Bouncer does not reciprocate this love.

B. Using the following symbolization key:

domain:	all animals
Dx:	______$_x$ is a dog
Sx:	______$_x$ likes samurai movies
Lxy:	______$_x$ is larger than ______$_y$
r:	Rave
h:	Shane
d:	Daisy

symbolize the following sentences in PL:

1. Rave is a dog who likes samurai movies.
2. Rave, Shane, and Daisy are all dogs.
3. Shane is larger than Rave, and Daisy is larger than Shane.
4. All dogs like samurai movies.
5. Only dogs like samurai movies.
6. There is a dog that is larger than Shane.
7. If there is a dog larger than Daisy, then there is a dog larger than Shane.
8. No animal that likes samurai movies is larger than Shane.
9. No dog is larger than Daisy.
10. Any animal that dislikes samurai movies is larger than Rave.
11. There is an animal that is between Rave and Shane in size.
12. There is no dog that is between Rave and Shane in size.
13. No dog is larger than itself.
14. Every dog is larger than some dog.
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.

C. Using the symbolization key given, symbolize each English-language sentence into PL.

domain:	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.

D. Using the following symbolization key:

domain:	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

symbolize the following English sentences in PL:

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.

E. Using the following symbolization key:

domain:	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

symbolize the following sentences in PL:

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 sons 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 brother who also dances ballet.
12. Every woman who dances ballet is the child of someone who dances ballet.

24 Sentences of PL

We know how to represent English sentences in PL. The time has finally come to define the notion of a *sentence* of PL.

24.1 Expressions

There are six kinds of symbols in PL:

Predicates For every $n \geqslant 0$, n-place predicates, $A, B, C, \ldots, Z$, or with subscripts, as needed: $A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$

Names $a, b, c, \ldots, t, u, v$, or with subscripts, as needed $a_1, b_{224}, h_7, u_{32}, \ldots$

Variables w, x, y, z, or with subscripts, as needed $w_1, x_1, y_1, z_1, w_2, \ldots$

Connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Brackets (,)

Quantifiers $\forall, \exists$

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

24.2 Sentences

We start by defining the notion of a term.

A TERM is any name or any variable.

So, here are some terms:

$$a, b, x, x_1 x_2, y, y_{254}, z$$

Next we need to define atomic sentences.

1. Any sentence letter is an atomic sentence.
2. If $\mathscr{R}$ is an n-place predicate and $t_1, t_2, \ldots, t_n$ are n terms, then $\mathscr{R}t_1t_2\ldots t_n$ is an atomic sentence.
3. Nothing else is an atomic sentence.

Note that the sentence letters also sentences of PL. In fact, clause (1) is not, strictly speaking, necessary. For we may understand sentence letters as 0-place predicates. An n-place predicate must be followed by n terms in order to form an atomic sentence; so, if 'A' is a 0-place predicate, then it must be followed by no terms in order to form an atomic sentence. So 'A' is an atomic sentence. So every atomic sentence of SL is also an atomic sentence of of PL.

The use of script letters here follows the conventions laid down in §7. So, '$\mathscr{R}$' is not itself a predicate of PL. Rather, it is a symbol of our metalanguage (augmented English) that we use to talk about any predicate of PL. Similarly, 't_1' is not a term of PL, but a symbol of the metalanguage that we can use to talk about any term of PL. So, where 'D' is a zero-place predicate, 'F' is a one-place predicate, 'G' is a three-place predicate, and 'S' is a six-place predicate, here are some atomic sentences:

$$D$$
$$Fx$$
$$Fa$$
$$Gxay$$
$$Sby_{254}zaaz$$

Once we know what atomic sentences are, we can offer recursion clauses to define arbitrary sentences. The first few clauses are exactly the same as for SL.

1. Every atomic sentence is a sentence.
2. If $\mathscr{A}$ is a sentence, then $\neg\mathscr{A}$ is a sentence.
3. If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \wedge \mathscr{B})$ is a sentence.
4. If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \vee \mathscr{B})$ is a sentence.
5. If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \rightarrow \mathscr{B})$ is a sentence.
6. If $\mathscr{A}$ and $\mathscr{B}$ are sentences, then $(\mathscr{A} \leftrightarrow \mathscr{B})$ is a sentence.
7. If $\mathscr{A}$ is a sentence and x is a variable, then $\forall x\, \mathscr{A}$ is a sentence.
8. If $\mathscr{A}$ is a sentence and x is a variable, then $\exists x\, \mathscr{A}$ is a sentence.
9. Nothing else is a sentence.

So, assuming again that 'F' is a one-place predicate, 'G' is a three-place predicate and 'H' is a six place-predicate, here are some sentences:

$$\begin{gathered} Fx \\ Gayz \\ Syzyayx \\ (Gayz \to Syzyayx) \\ \forall z(Gayz \to Syzyayx) \\ Fx \leftrightarrow \forall z(Gayz \to Syzyayx) \\ \exists y(Fx \leftrightarrow \forall z(Gayz \to Syzyayx)) \\ \forall x\exists y(Fx \leftrightarrow \forall z(Gayz \to Syzyayx)) \end{gathered}$$

We can now give a formal definition of scope, which incorporates the definition of the scope of a quantifier. Here we follow the case of SL, though we note that a logical operator can be either a connective or a quantifier:

The MAIN OPERATOR in a sentence is the operator that was introduced last, when that sentence was constructed using the rules for sentences.

The SCOPE of an operator (in a sentence) is the sub-sentence for which that operator is the main operator.

So we can graphically illustrate the scope of the quantifiers in the preceding example thus:

$$\overbrace{\forall x\ \overbrace{\exists y\ (Fx \leftrightarrow \overbrace{\forall z\ (Gayz \to Syzyayx)}^{\text{scope of '}\forall z\text{'}})}^{\text{scope of '}\exists y\text{'}}}^{\text{scope of '}\forall x\text{'}}$$

Associated with each quantifier in a sentence is a variable. For instance, in the sentence '$\forall x\, Fx$', the variable 'x' is associated with the universal quantifier '$\forall$'. For this reason, let's call '$\forall x$' an x-quantifier. Likewise, in the sentence '$\exists w\,(Gw \to Hwa)$', the variable 'w' is associated with the existential quantifier '$\exists$'. So we'll call '$\exists w$' a w-quantifier.

If a variable, x, falls within the scope of an x-quantifier, then we will say that that variable is *bound*. If a variable, x, does not fall within the scope of any x-quantifier, then we will will say that that variable is *free*.

A BOUND VARIABLE is an occurrence of a variable x that is within the scope of either $\forall x$ or $\exists x$.

A FREE VARIABLE is any occurrence of a variable that is not bound.

For example, in the sentence 'Fx', 'x' is *free*. In the sentence '$\forall x \forall y Fxy$', both 'x' and 'y' are *bound*. In '$\forall x Px \rightarrow Qx$', the '$x$' in '$Px$' is bound, whereas the 'x' in 'Qx' is free. Or consider the sentence

$$\forall x(Ex \lor Dy) \rightarrow \exists z(Ex \rightarrow Lzx)$$

The scope of the universal quantifier '$\forall x$' is '$\forall x(Ex \lor Dy)$', so the first 'x' after '$\forall x$' is bound by the universal quantifier. However, the second and third occurrences of 'x' after '$\forall x$' are free. Similarly, the 'y' is free. The scope of the existential quantifier '$\exists z$' is '$\exists z\,(Ex \rightarrow Lzx)$', so the '$z$' in '$Lzx$' is bound.

What about the variable which immediately follows the quantifier? For instance, in the sentence '$\forall x\, Jx$', there are two occurrences of 'x'. The second one is bound—but what about the first? It doesn't much matter what we say about this variable, but it follows from our definition above that it is will also count as bound (for it occurs within the scope of '$\forall x$'). Still, since these occurrences of the variable are part of the quantifiers, it makes more sense to think of them as the variables which *do* the binding, rather than those which end up being bound.

A single bound variable, $\mathscr{x}$, may fall within the scope of multiple $\mathscr{x}$-quantifiers. For instance, consider the sentence

$$\forall w\,(\exists y\, Lwy \rightarrow \exists w\, Aw)$$

The final 'w' (in 'Aw') both occurs within the scope of the universal quantifier '$\forall w$' and within the scope of the existential quantifier '$\exists w$'. This occurrence of 'w' is bound—but which quantifier binds it? We will say that it is bound by the *existential* quantifier. To see why, think about how we would build the sentence up according to the rules for sentences. First, we would show that 'Aw' was a sentence (because it's a atomic sentence). In this sentence, the variable 'w' appears *free*. Next, we would appeal to the rules for quantifiers to establish that '$\exists w\, Aw$' is a sentence. In this sentence, the variable w is bound. So adding the quantifier $\exists w$ changed the w in 'Aw' from free to bound. For this reason, we will say that the existential quantifier '$\exists w$' *binds* the w in 'Aw'. In general, in order to see which variables a quantifier binds, consider the sub-sentence for which that quantifier is the main operator: '$\forall \mathscr{x} \mathscr{A}$' or '$\exists \mathscr{x}\, \mathscr{A}$'. Then, remove the quantifier from this sub-sentence, leaving behind just the sub-sentence '$\mathscr{A}$'. If an occurrence of $\mathscr{x}$ appears *free* in '$\mathscr{A}$', then this occurrence is *bound* by the quantifier.

In a PL sentence of the form $\forall \mathscr{x} \mathscr{A}$, the quantifier $\forall \mathscr{x}$ binds every *free* occurrence of $\mathscr{x}$ in $\mathscr{A}$. If an occurrence of $\mathscr{x}$ in $\mathscr{A}$ is already bound, then the quantifier $\forall \mathscr{x}$ does not bind it.

Similarly,

In a PL sentence of the form $\exists \mathscr{x} \mathscr{A}$, the quantifier $\exists \mathscr{x}$ binds every *free* occurrence of $\mathscr{x}$ in $\mathscr{A}$. If an occurrence of $\mathscr{x}$ in $\mathscr{A}$ is already bound, then the quantifier $\exists \mathscr{x}$ does not bind it.

Finally: if a sentence of PL contains an a free variable, then we say that that sentence is *open*. And, if a sentence of PL contains no free variables, then we say that that sentence is *closed*.

When we turn to the semantics of PL in part VI, we'll see that the free variables in an open sentence function as names. However, conceptually, we will want to keep names and variables separate. So, when translating from English to PL, we should avoid open sentences. Open sentences contain variables masquerading as names—better to be up front and just use names. So, instead of translating 'Adam loves himself' with 'Lxx', you should instead translate it with 'Laa'—even though, technically, the first expression is a sentence of PL, and it could even be synonymous with the second. (Some texts introduce more complicated rules for when expressions are grammatical sentences with the aim of getting open sentences to come out as ungrammatical. Here, I've opted for a simpler syntax. This simpler syntax counts 'Fx' as grammatical, but this is harmless, as this sentence will end up being perfectly meaningful—the free 'x' will refer to something, just as a name would. However, we will avoid confusion by resolving to always translate names from English with the names of PL.)

24.3 Parentheses conventions

We will adopt the same notational conventions governing parentheses that we did for SL (see §6 and §10.3.)

First, we may omit the outermost parentheses of a formula.

Second, we may use square parentheses, '[' and ']', in place of parentheses to increase the readability of formulas.

Practice exercises

A. Identify which variables are bound and which are free.

1. $\exists x\, Lx, y \land \forall y\, Lyx$
2. $\forall x\, Ax \land Bx$
3. $\forall x(Ax \land Bx) \land \forall y(Cx \land Dy)$
4. $\forall x \exists y[Rxy \to (Jz \land Kx)] \lor Ryx$
5. $\forall x_1(Mx_2 \leftrightarrow Lx_2x_1) \land \exists x_2\, Lx_3x_2$

PART VI

Semantics for Predicate Logic

25 | Extensionality

Recall that SL is a truth-functional language. Its operators are all truth-functional, and *all* that we can do with SL is key sentences to particular truth values. We can do this *directly*. For example, we might stipulate that the SL sentence 'P' is to be true. Alternatively, we can do this *indirectly*, offering a symbolization key, e.g.:

P: Big Ben is in London

Now recall from §9 that this should be taken to mean:

- The SL sentence 'P' is to take the same truth value as the English sentence 'Big Ben is in London' (whatever that truth value may be)

The point that we emphasized is that SL cannot handle differences in meaning that go beyond mere differences in truth value.

25.1 Symbolizing versus translating

PL has some similar limitations, but it goes beyond mere truth values, since it enables us to split up sentences into terms, predicates and quantifier expressions. This enables us to consider what is *true of* some particular object, or of some or all objects. But we can do no more than that.

When we provide a symbolization key for some PL predicates, such as:

Cx: $\_\_\_\_\__x$ teaches Logic III in Calgary

we do not carry the *meaning* of the English predicate across into our PL predicate. We are simply stipulating something like the following:

- 'Cx' and '$\_\_\_\_\__x$ teaches Logic III in Calgary' are to be *true of* exactly the same things.

So, in particular:

- 'Cx' is to be true of all and only those things which teach Logic III in Calgary (whatever those things might be).

This is an indirect stipulation. Alternatively, we can directly stipulate which objects a predicate should be true of. For example, we can stipulate that 'Cx' is to be true of Richard Zach, and Richard Zach alone. As it happens, this direct stipulation would have the same effect as the indirect stipulation. Note, however, that the English predicates '_____ is Richard Zach' and '_____ teaches Logic III in Calgary' have very different meanings!

The point is that PL does not give us any resources for dealing with nuances of meaning. When we interpret PL, all we are considering is what the predicates are true of, regardless of whether we specify these things directly or indirectly. The things a predicate is true of are known as the EXTENSION of that predicate. We say that PL is an EXTENSIONAL LANGUAGE because PL does not represent differences of meaning between predicates that have the same extension.

For this reason, we say only that PL sentences *symbolize* English sentences. It is doubtful that we are *translating* English into PL, as translations should preserve meanings, and not just extensions.

25.2 A word on extensions

We can stipulate directly what predicates are to be true of, so it is worth noting that our stipulations can be as arbitrary as we like. For example, we could stipulate that 'Hx' should be true of, and only of, the following objects:

Justin Trudeau
the number π
every top-F key on every piano ever made

Now, the objects that we have listed have nothing particularly in common. But this doesn't matter. Logic doesn't care about what strikes us mere humans as 'natural' or 'similar'. Armed with this interpretation of 'Hx', suppose we now add to our symbolization key:

j:	Justin Trudeau
r:	Rachel Notley
p:	the number π

Then 'Hj' and 'Hp' will both be true, on this interpretation, but 'Hr' will be false, since Rachel Notley was not among the stipulated objects.

25.3 Many-place predicates

All of this is quite easy to understand when it comes to one-place predicates, but it gets messier when we consider two-place predicates. Consider a symbolization key like:

Lxy: $\_\_\_\_\__x$ loves $\_\_\_\_\__y$

Given what we said above, this symbolization key should be read as saying:

- 'Lxy' and '$\_\_\_\_\__x$ loves $\_\_\_\_\__y$' are to be true of exactly the same things

So, in particular:

- 'Lxy' is to be true of x and y (in that order) iff x loves y.

It is important that we insist upon the order here, since love—famously—is not always reciprocated. (Note that 'x' and 'y' on the right here are symbols of augmented English, and that they are being *used*. By contrast, 'x' and 'y' in 'Lxy' are symbols of PL, and they are being *mentioned*.)

That is an indirect stipulation. What about a direct stipulation? This is slightly harder. If we *simply* list objects that fall under 'Lxy', we will not know whether they are the lover or the beloved (or both). We have to find a way to include the order in our explicit stipulation.

To do this, we can specify that two-place predicates are true of *pairs* of objects, where the order of the pair is important. Thus we might stipulate that 'Bxy' is to be true of, and only of, the following pairs of objects:

⟨Lenin, Marx⟩
⟨Heidegger, Sartre⟩
⟨Sartre, Heidegger⟩

Here the angle-brackets keep us informed concerning order. Suppose we now add the following stipulations:

l:	Lenin
m:	Marx
h:	Heidegger
r:	Sartre

Then 'Blm' will be true, since ⟨Lenin, Marx⟩ was in our explicit list, but 'Bml' will be false, since ⟨Marx, Lenin⟩ was not in our list. However, both 'Bhr' and 'Brh' will be true, since both ⟨Heidegger, Sartre⟩ and ⟨Sartre, Heidegger⟩ are in our explicit list.

To make these ideas more precise, we would need to develop some *set theory*. That would give us some precise tools for dealing with extensions and with ordered pairs (and ordered triples, etc.). However, set theory is not covered in this book, so we will leave these ideas at an imprecise level. Nevertheless, the general idea should be clear.

25.4 Interpretation

We defined a VALUATION in SL as any assignment of truth and falsity to sentence letters. In PL, we are going to define an INTERPRETATION as consisting of four things:

- the specification of a domain
- for each sentence letter we care to consider, a truth value

- for each name or free variable that we care to consider, an assignment of exactly one object within the domain
- for each predicate that we care to consider, a specification of what things (in what order) the predicate is to be true of

The symbolization keys that we considered in Part V consequently give us one very convenient way to present an interpretation. We will continue to use them throughout this chapter. However, it is sometimes also convenient to present an interpretation *diagrammatically*.

Suppose we want to consider just a single two-place predicate, 'Rxy'. Then we can represent it just by drawing an arrow between two objects, and stipulate that 'Rxy' is to hold of x and y just in case there is an arrow running from x to y in our diagram. As an example, we might offer:

$$\begin{tikzcd}
1 \arrow[r] \arrow[dr] & 2 \arrow[d] \\
4 \arrow[u] & 3 \arrow[l]
\end{tikzcd}$$

This would be suitable to characterize an interpretation whose domain is the first four positive whole numbers, and which interprets 'Rxy' as being true of and only of:

$$\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle, \langle 4, 1\rangle, \langle 1, 3\rangle$$

Equally we might offer:

$$\begin{tikzcd}
1 \arrow[loop left] \arrow[dr, leftrightarrow] & 2 \\
4 & 3 \arrow[l] \arrow[loop right]
\end{tikzcd}$$

for an interpretation with the same domain, which interprets 'Rxy' as being true of and only of:

$$\langle 1, 3\rangle, \langle 3, 1\rangle, \langle 3, 4\rangle, \langle 1, 1\rangle, \langle 3, 3\rangle$$

If we wanted, we could make our diagrams more complex. For example, we could add names as labels for particular objects. Equally, to symbolize the extension of a one-place predicate, we might simply draw a ring around some particular objects and stipulate that the thus encircled objects (and only them) are to fall under the predicate 'Hx', say.

26 | Truth in PL

We know what interpretations are. Since, among other things, they tell us which predicates are true of which objects, they will provide us with an account of the truth of atomic sentences. However, we must also present a detailed account of what it is for an arbitrary PL sentence to be true or false in an interpretation.

We know from §24 that there are three kinds of sentence in PL:

- atomic sentences
- sentences whose main operator is a sentential operator
- sentences whose main operator is a quantifier

We need to explain truth for all three kinds of sentence.

We will provide a completely general explanation in this section. However, to try to keep the explanation comprehensible, we will, at several points, use the following interpretation:

domain:	all people born before 2000CE
a:	Aristotle
b:	Beyoncé
Px:	$\_\_\_\_\__x$ is a philosopher
Rxy:	$\_\_\_\_\__x$ was born before $\_\_\_\_\__y$

This will be our *go-to example* in what follows.

26.1 Atomic sentences

The truth of atomic sentences should be fairly straightforward. For sentence letters, the interpretation specifies if it is true or false. The sentence 'Pa' should be true just in case 'Px' is true of 'a'. Given our go-to interpretation, this is true iff Aristotle is a philosopher. Aristotle is a philosopher. So the sentence is true. Equally, 'Pb' is false on our go-to interpretation.

Likewise, on this interpretation, 'Rab' is true iff the object named by 'a' was born before the object named by 'b'. Well, Aristotle was born before Beyoncé. So 'Rab' is true. Equally, 'Raa' is false: Aristotle was not born before Aristotle.

Dealing with atomic sentences, then, is very intuitive. When $\mathscr{R}$ is an n-place predicate and $t_1, t_2, \ldots, t_n$ are n terms,

$\mathscr{R}t_1t_2\ldots t_n$ is true in an interpretation **iff**
$\mathscr{R}$ is true of the objects named by $t_1, t_2, \ldots, t_n$ in that interpretation (considered in that order)

26.2 Sentential operators

We saw in §24 that PL sentences can be built up from simpler ones using the truth-functional operators that were familiar from SL. The rules governing these truth-functional operators are *exactly* the same as they were when we considered SL. Here they are:

$\mathscr{A} \wedge \mathscr{B}$ is true in an interpretation **iff**
both $\mathscr{A}$ is true and $\mathscr{B}$ is true in that interpretation

$\mathscr{A} \vee \mathscr{B}$ is true in an interpretation **iff**
either $\mathscr{A}$ is true or $\mathscr{B}$ is true in that interpretation

$\neg\mathscr{A}$ is true in an interpretation **iff**
$\mathscr{A}$ is false in that interpretation

$\mathscr{A} \rightarrow \mathscr{B}$ is true in an interpretation **iff**
either $\mathscr{A}$ is false or $\mathscr{B}$ is true in that interpretation

$\mathscr{A} \leftrightarrow \mathscr{B}$ is true in an interpretation **iff**
$\mathscr{A}$ has the same truth value as $\mathscr{B}$ in that interpretation

This presents the very same information as the characteristic truth tables for the operators; it just does so in a slightly different way. Some examples will probably help to illustrate the idea. On our go-to interpretation:

- 'Pa' is true
- '$Rab \wedge Pb$' is false because, although 'Rab' is true, 'Pb' is false
- '$Pa \vee Pb$' is true
- 'Rba' is false
- '$Pa \wedge \neg(Pb \wedge Ra, b)$' is true, because '$Pa$' is true and '$Pb$' is false

Make sure you understand these examples.

26.3 When the main operator is a quantifier

The exciting innovation in PL, though, is the use of *quantifiers*, but expressing the truth conditions for quantified sentences is a bit more fiddly than one might first expect.

We want to be able to say that '$\forall x \exists y\, Lxy$' is true just in case '$\exists y\, Lxy$' is true of everything in the domain. This is problematic, since our interpretation does not directly specify what '$\exists y\, Lxy$' is to be true of. Instead, whether or not this is true of something should follow just from the interpretation of 'Lxy', the domain, and the meanings of the quantifiers.

Here is a naïve thought. We might try to say that '$\forall x \exists y\, Lxy$' is to be true in an interpretation iff $\exists y\, L\mathscr{a}y$ is true for *every* name $\mathscr{a}$ that we have included in our interpretation. Similarly, we might try to say that $\exists y\, L\mathscr{a}y$ is true just in case $L\mathscr{a}\mathscr{b}$ is true for *some* name $\mathscr{b}$ that we have included in our interpretation.

Unfortunately, this is not right. To see this, observe that in our go-to interpretation, we have only given interpretations for *two* names, 'a' and 'b', but the domain—all people born before the year 2000CE—contains many more than two people. We have no intention of trying to name *all* of them!

So here is another thought. (And this thought is not naïve, but correct.) Although it is not the case that we have named *everyone*, each person *could* have been given a name—in fact, every such person could have been named by the variable, 'x'. So we should focus on this possibility of modifying our interpretation, by making x into a name for someone in the domain. We will offer a few examples of how this might work, centering on our go-to interpretation, and we will then present the formal definition.

In our go-to interpretation, '$\exists x\, Rbx$' should be true. After all, in the domain, there is certainly someone who was born after Beyoncé. Lady Gaga is one of those people. Indeed, if we were to modify our go-to interpretation—temporarily, mind—by having 'x' refer to Lady Gaga, then 'Rbx' would be true on this modified interpretation. And this will suffice to make '$\exists x\, Rbx$' true on the original go-to interpretation.

In our go-to interpretation, '$\exists x(Px \wedge Rxa)$' should also be true. After all, in the domain, there is certainly someone who was both a philosopher and born before Aristotle. Socrates is one such person. Indeed, if we were to modify our go-to interpretation by letting 'x' denote Socrates, then '$Px \wedge Rxa$' would be true on this modified interpretation. Again, this will suffice to make '$\exists x(Px \wedge Rxa)$' true on the original interpretation.

In our go-to interpretation, '$\forall x\, \exists y\, Rxy$' should be false. After all, consider the last person born in the year 1999. We don't know who that was, but if we were to modify our go-to interpretation by letting 'x' denote that person, then we would not be able to find anyone else in the domain to denote with 'y' in such a way that 'Rxy' would be true. Indeed, no matter *whom* we named with 'y', 'Rxy' would be false. This observation is sufficient to make '$\exists y\, Rxy$' *false* in our modified interpretation, which in turn is sufficient to make '$\forall x \exists y\, Rxy$' false on the original interpretation.

If you have understood these three examples, good. That's what matters. Strictly speaking, though, we still need to give a precise definition of the truth conditions for quantified sentences. The result, sadly, is a bit ugly, and requires a few new definitions. Brace yourself!

Our interpretation will include a specification of which objects are in the domain. If our interpretation is **I**, and d is some thing in the domain of **I**, then we can consider the *modified* interpretation $\mathbf{I}[x \to d]$ which is just like **I** except that, in it, x names d.

If **I** is an interpretation, d is an object in the domain of **I**, and $\mathcal{x}$ is a variable, then the modified interpretation $\mathbf{I}[\mathcal{x} \rightarrow d]$ is an interpretation which is exactly like **I**, except that the variable $\mathcal{x}$ names d.

So, for instance, if **I** is our go-to interpretation, then $\mathbf{I}[z \rightarrow \text{Socrates}]$ is the interpretation:

domain:	all people born before 2000CE
a:	Aristotle
b:	Beyoncé
z:	Socrates
Px:	$\_\_\_\_\__x$ is a philosopher
Rxy:	$\_\_\_\_\__x$ was born before $\_\_\_\_\__y$

Notice that 'Pz' is true in this (modified) interpretation. For this reason, we will say that '$\exists z\, Pz$' is true in the original, go-to interpretation.

For another example, if **I** is our go-to interpretation, then $\mathbf{I}[x \rightarrow \text{Beyoncé}]$ is the interpretation:

domain:	all people born before 2000CE
a:	Aristotle
b:	Beyoncé
x:	Beyoncé
Px:	$\_\_\_\_\__x$ is a philosopher
Rxy:	$\_\_\_\_\__x$ was born before $\_\_\_\_\__y$

Notice that 'Px' is false in this (modified) interpretation. For this reason, we will say that '$\forall x\, Px$' is false in the original, go-to interpretation.

More generally, we will say that a sentence of the form $\exists \mathcal{x}\, \mathcal{A}$ is true in an interpretation **I** if and only if, for *some* d in the domain of **I**, $\mathcal{A}$ is true in the modified interpretation $\mathbf{I}[\mathcal{x} \rightarrow d]$, in which the variable $\mathcal{x}$ is a name for d.

$\exists \mathcal{x}\, \mathcal{A}$ is true in an interpretation **I** iff
$\mathcal{A}$ is true in the modified interpretation $\mathbf{I}[\mathcal{x} \rightarrow d]$, for some d in the domain.

Similarly, we will say that a sentence of the form $\forall \mathcal{x}\, \mathcal{A}$ is true in an interpretation **I** if and only if, for *every* d in the domain of **I**, $\mathcal{A}$ is true in the modified interpretation $\mathbf{I}[\mathcal{x} \rightarrow d]$, in which the variable $\mathcal{x}$ is a name for d.

$\forall \mathcal{x} \mathcal{A}$ is true in an interpretation **I** iff
$\mathcal{A}$ is true in the modified interpretation $\mathbf{I}[\mathcal{x} \rightarrow d]$, for every d in the domain.

To be clear: all this is doing is formalizing (very pedantically) the intuitive idea expressed on the previous page. The result is a bit ugly, and the final definition might look a bit opaque. Hopefully, though, the *spirit* of the idea is clear.

Note that all of the foregoing extends to sentences involving multiple quantifiers. Consider, for instance, the sentence '$\exists x\, \exists y\, Rxy$'. To determine whether this is true, we should see whether there is anything in the domain such that, when we allow 'x' to name that thing, '$\exists y\, Rxy$' is true. There is. '$\exists y\, Rxy$' says there is someone such that x was born before them—or, in other words, that x was born before someone. If x is a name for Socrates, then this will be true. So, consider the modified interpretation $\mathbf{I}[x \rightarrow \text{Socrates}]$:

domain:	all people born before 2000CE
a:	Aristotle
b:	Beyoncé
x:	Socrates
Px:	$\_\_\_\_\__x$ is a philosopher
Rxy:	$\_\_\_\_\__x$ was born before $\_\_\_\_\__y$

In order to check that the sentence '$\exists y\, Rxy$' is true in the interpretation $\mathbf{I}[x \rightarrow \text{Socrates}]$, we should consider whether there is anything in the domain such that, if we allow y to name that thing, the sentence 'Rxy' will be true. There is—just let y name Lady Gaga. Since Socrates was born before Lady Gaga, 'Rxy' will be true. That is, consider the modified interpretation: $\mathbf{I}[x \rightarrow \text{Socrates}, y \rightarrow \text{Lady Gaga}]$:

domain:	all people born before 2000CE
a:	Aristotle
b:	Beyoncé
x:	Socrates
y:	Lady Gaga
Px:	$\_\_\_\_\__x$ is a philosopher
Rxy:	$\_\_\_\_\__x$ was born before $\_\_\_\_\__y$

Since Socrates was born before Lady Gaga, 'Rxy' is true in the interpretation $\mathbf{I}[x \rightarrow \text{Socrates}, y \rightarrow \text{Lady Gaga}]$. So '$\exists y\, Rxy$' is true in the interpretation $\mathbf{I}[x \rightarrow \text{Socrates}]$. So '$\exists x\, \exists y\, Rxy$' is true in the go-too interpretation $\mathbf{I}$.

Practice exercises

A. Consider the following interpretation:

- The domain comprises only Corwin and Benedict
- 'Ax' is to be true of both Corwin and Benedict
- 'Bx' is to be true of Benedict only
- 'Nx' is to be true of no one
- 'c' is to refer to Corwin

Determine whether each of the following sentences is true or false in that interpretation:

1. Bc
2. $Ac \leftrightarrow \neg Nc$
3. $Nc \rightarrow (Ac \lor Bc)$
4. $\forall x\, Ax$
5. $\forall x \neg Bx$
6. $\exists x(Ax \land Bx)$
7. $\exists x(Ax \rightarrow Nx)$
8. $\forall x(Nx \lor \neg Nx)$
9. $\exists x\, Bx \rightarrow \forall x\, Ax$

B. Consider the following interpretation:

- The domain comprises only Lemmy, Courtney and Eddy
- 'Gx' is to be true of Lemmy, Courtney and Eddy.
- 'Hx' is to be true of and only of Courtney
- 'Mx' is to be true of and only of Lemmy and Eddy
- 'c' is to refer to Courtney
- 'e' is to refer to Eddy

Determine whether each of the following sentences is true or false in that interpretation:

1. Hc
2. He
3. $Mc \lor Me$
4. $Gc \lor \neg Gc$
5. $Mc \rightarrow Gc$
6. $\exists x\, Hx$
7. $\forall x\, Hx$
8. $\exists x \neg Mx$
9. $\exists x(Hx \land Gx)$
10. $\exists x(Mx \land Gx)$
11. $\forall x(Hx \lor Mx)$
12. $\exists x\, Hx \land \exists x\, Mx$
13. $\forall x(Hx \leftrightarrow \neg Mx)$
14. $\exists x\, Gx \land \exists x \neg Gx$
15. $\forall x \exists y(Gx \land Hy)$

C. Following the diagram conventions introduced at the end of §25, consider the following interpretation:

$$\begin{tikzcd}
1 \arrow[loop left] \arrow[r] \arrow[d] \arrow[dr, leftrightarrow] \arrow[drr] & 2 \arrow[dr] & \\
3 & 4 & 5 \arrow[loop right]
\end{tikzcd}$$

Determine whether each of the following sentences is true or false in that interpretation:

1. $\exists x\, Rxx$
2. $\forall x\, Rxx$
3. $\exists x \forall y\, Rxy$
4. $\exists x \forall y\, Ryx$
5. $\forall x \forall y \forall z ((Rxy \wedge Ryz) \rightarrow Rxz)$
6. $\forall x \forall y \forall z ((Rxy \wedge Rxz) \rightarrow Ryz)$
7. $\exists x \forall y\, \neg Rxy$
8. $\forall x (\exists y\, Rxy \rightarrow \exists y\, Ryx)$

27 Semantic concepts

Offering a precise definition of truth in PL was more than a little fiddly, but now that we are done, we can define various central logical notions. These will look very similar to the definitions we offered for SL. However, remember that they concern *interpretations*, rather than valuations.

We will use the symbol '$\vDash$' for PL much as we did for SL. So:

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vDash \mathscr{C}$$

means that there is no interpretation in which all of $\mathscr{A}_1$, $\mathscr{A}_2$, …, $\mathscr{A}_n$ are true and in which $\mathscr{C}$ is false. Derivatively,

$$\vDash \mathscr{A}$$

means that there is no interpretation in which $\mathscr{A}$ is false; so it means that $\mathscr{A}$ is true in every interpretation (so: $\mathscr{A}$ is a tautology). And

$$\mathscr{A} \vDash$$

means that there is no interpretation in which $\mathscr{A}$ is true; so it means that $\mathscr{A}$ is false in every interpretation (so: $\mathscr{A}$ is a contradiction).

The other logical notions also have corresponding definitions in PL:

- An PL sentence $\mathscr{A}$ is a TAUTOLOGY (in PL) iff $\mathscr{A}$ is true in every interpretation; i.e., $\vDash \mathscr{A}$.

- $\mathscr{A}$ is a CONTRADICTION (in PL) iff $\mathscr{A}$ is false in every interpretation; i.e., $\vDash \neg\mathscr{A}$.

- $\mathscr{A}_1, \mathscr{A}_2, \ldots \mathscr{A}_n \therefore \mathscr{C}$ is VALID (in PL) iff there is no interpretation in which all of the premises are true and the conclusion is false; i.e., $\mathscr{A}_1, \mathscr{A}_2, \ldots \mathscr{A}_n \vDash \mathscr{C}$. It is INVALID IN PL otherwise.

- Two PL sentences $\mathscr{A}$ and $\mathscr{B}$ are EQUIVALENT (in PL) iff they are true in exactly the same interpretations as each other; i.e., both $\mathscr{A} \vDash \mathscr{B}$ and $\mathscr{B} \vDash \mathscr{A}$.

- The PL sentences $\mathscr{A}_1$, $\mathscr{A}_2$, …, $\mathscr{A}_n$ are SATISFIABLE (in PL) iff there is some interpretation in which all of the sentences are true. They are UNSATISFIABLE (in PL) iff there is no such interpretation.

28 | Using interpretations

28.1 Tautologies and contradictions

Suppose we want to show that '$\exists x\, Axx \rightarrow Bd$' is *not* a tautology. This requires showing that the sentence is not true in every interpretation; i.e., that it is false in some interpretation. If we can provide just one interpretation in which the sentence is false, then we will have shown that the sentence is not a tautology.

In order for '$\exists x\, Axx \rightarrow Bd$' to be false, the antecedent ('$\exists x\, Axx$') must be true, and the consequent ('Bd') must be false. To construct such an interpretation, we start by specifying a domain. Keeping the domain small makes it easier to specify what the predicates will be true of, so we will start with a domain that has just one member. For concreteness, let's say it is the city of Paris.

domain: Paris

The name 'd' must refer to something in the domain, so we have no option but:

d: Paris

Recall that we want '$\exists x\, Axx$' to be true, so we want all members of the domain to be paired with themselves in the extension of 'A'. We can just offer:

Axy: $\_\_\_\_\__x$ is identical with $\_\_\_\_\__y$

Now 'Add' is true, so it is surely true that '$\exists x\, Axx$'. Next, we want 'Bd' to be false, so the referent of 'd' must not be in the extension of 'B'. We might simply offer:

Bx: $\_\_\_\_\__x$ is in Germany

Now we have an interpretation where '$\exists x\, Axx$' is true, but where 'Bd' is false. So there is an interpretation where '$\exists x\, Axx \rightarrow Bd$' is false. So '$\exists x\, Axx \rightarrow Bd$' is not a tautology.

We can just as easily show that '$\exists x Axx \rightarrow Bd$' is not a contradiction. We need only specify an interpretation in which '$\exists x Axx \rightarrow Bd$' is true; i.e., an interpretation in which either '$\exists x\, Axx$' is false or 'Bd' is true. Here is one:

domain: Paris
d: Paris

Axy: _____$_x$ is identical with _____$_y$
Bx: _____$_x$ is in France

This shows that there is an interpretation where '$\exists x Axx \rightarrow Bd$' is true. So '$\exists x\, Axx \rightarrow Bd$' is not a contradiction.

28.2 Equivalence

Suppose we want to show that '$\forall x\, Sx$' and '$\exists x\, Sx$' are not equivalent. We need to construct an interpretation 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 domain. Again, we make the domain small so that we can specify extensions easily. In this case, we will need at least two objects. (If we chose a domain with only one member, the two sentences would end up with the same truth value. In order to see why, try constructing some partial interpretations with one-member domains.) For concreteness, let's take:

domain: Ornette Coleman, Miles Davis

We can make '$\exists x\, Sx$' true by including something in the extension of 'S', and we can make '$\forall x\, Sx$' false by leaving something out of the extension of 'S'. For concreteness we will offer:

Sx: _____$_x$ plays saxophone

Now '$\exists x\, Sx$' is true, because 'Sx' is true of Ornette Coleman. Slightly more precisely, modify our interpretation by allowing 'x' to name Ornette Coleman. 'Sx' is true in this modified interpretation, so '$\exists x\, Sx$' was true in the original interpretation. Similarly, '$\forall x\, Sx$' is false, because 'Sx' is false of Miles Davis. Slightly more precisely, modify our interpretation by allowing 'x' to name Miles Davis, and 'Sx' is false in this modified interpretation, so '$\forall x\, Sx$' was false in the original interpretation. We have provided a counter-interpretation to the claim that '$\forall x\, Sx$' and '$\exists x\, Sx$' are equivalent.

To show that $\mathscr{A}$ is not a tautology, it suffices to find an interpretation where $\mathscr{A}$ is false.
To show that $\mathscr{A}$ is not a contradiction, it suffices to find an interpretation where $\mathscr{A}$ is true.
To show that $\mathscr{A}$ and $\mathscr{B}$ are not equivalent, it suffices to find an interpretation where one is true and the other is false.

28.3 Validity, entailment and satisfiability

To test for validity, entailment, or satisfiability, we typically need to produce interpretations that determine the truth value of several sentences simultaneously.

Consider the following argument in PL:

$$\exists x(Gx \rightarrow Ga) \therefore \exists x\, Gx \rightarrow Ga$$

To show that this is invalid, we must make the premise true and the conclusion false. The conclusion is a conditional, so to make it false, the antecedent must be true and the consequent must be false. Clearly, our domain must contain two objects. Let's try:

domain: Karl Marx, Ludwig von Mises
Gx: $\underline{\quad\quad}_x$ hated communism
a: Karl Marx

Given that Marx wrote *The Communist Manifesto*, 'Ga' is plainly false in this interpretation. But von Mises famously hated communism, so '$\exists x\, Gx$' is true in this interpretation. Hence '$\exists x\, Gx \rightarrow Ga$' is false, as required.

Does this interpretation make the premise true? Yes it does! Note that '$Ga \rightarrow Ga$' is true. (Indeed, it is a tautology.) But then certainly '$\exists x(Gx \rightarrow Ga)$' is true, so the premise is true, and the conclusion is false, in this interpretation. The argument is therefore invalid.

In passing, note that we have also shown that '$\exists x(Gx \rightarrow Ga)$' does *not* entail '$\exists x\, Gx \rightarrow Ga$'. Equally, we have shown that the sentences '$\exists x(Gx \rightarrow Ga)$' and '$\neg(\exists x\, Gx \rightarrow Ga)$' are jointly satisfiable.

Let's consider a second example. Consider:

$$\forall x \exists y\, Lxy \therefore \exists y \forall x\, Lxy$$

Again, we want to show that this is invalid. To do this, we must make the premises true and the conclusion false. Here is a suggestion:

domain: UK citizens currently in a civil partnership with another UK citizen
Lxy: $\underline{\quad\quad}_x$ is in a civil partnership with $\underline{\quad\quad}_y$

The premise is clearly true on this interpretation. Anyone in the domain is a UK citizen in a civil partnership with some other UK citizen. That other citizen will also, then, be in the domain. So for everyone in the domain, there will be someone (else) in the domain with whom they are in a civil partnership. Hence '$\forall x \exists y\, Lxy$' is true. However, the conclusion is clearly false, for that would require that there is some single person who is in a civil partnership with everyone in the domain, and there is no such person, so the argument is invalid. We observe immediately that the sentences '$\forall x \exists y\, Lxy$' and '$\neg\exists y \forall x\, Lxy$' are jointly satisfiable and that '$\forall x \exists y\, Lxy$' does not entail '$\exists y \forall x\, Lxy$'.

For our third example, we'll mix things up a bit. In §25, we described how we can present some interpretations using diagrams. For example:

$$\begin{tikzcd}
1 \arrow[loop left] \arrow[r] \arrow[dr] & 2 \arrow[d] \\
& 3
\end{tikzcd}$$

Using the conventions employed in §25, the domain of this interpretation is the first three positive whole numbers, and 'Rxy' is true of x and y just in case there is an arrow from x to y in our diagram. Here are some sentences that the interpretation makes true:

- '$\forall x \exists y\, Ryx$'
- '$\exists x \forall y\, Rxy$' witness 1
- '$\exists x \forall y\, \neg Rxy$' witness 3
- '$\exists x(\exists y\, Ryx \wedge \neg\exists y\, Rxy)$' witness 3

This immediately shows that all of the preceding four sentences are jointly satisfiable. We can use this observation to generate *invalid* arguments, e.g.:

$$\forall x \exists y\, Ryx, \exists x \forall y\, Rxy \therefore \neg\exists x \forall y\, \neg Rxy$$

and many more besides.

To show that $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \therefore \mathscr{C}$ is invalid, it suffices to find an interpretation where all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are true and where $\mathscr{C}$ is false.
That same interpretation will show that $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ do not entail $\mathscr{C}$.
It will also show that $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n, \neg\mathscr{C}$ are jointly satisfiable.

When you provide an interpretation to refute a claim—to show that a sentence is not a tautology, say, or that an entailment fails—this is sometimes called providing a *counter-interpretation* (or providing a *counter-model*).

Practice exercises

A. Show that each of the following is neither a tautology nor a contradiction:

1. $Da \wedge Db$
2. $\exists x\, Txh$
3. $Pm \wedge \neg\forall x\, Px$
4. $\forall z Jz \leftrightarrow \exists y\, Jy$
5. $\forall x(Wxmn \vee \exists y Lxy)$
6. $\exists x(Gx \rightarrow \forall y\, My)$

B. Show that the following pairs of sentences are not equivalent.

1. Ja, Ka
2. $\exists x\, Jx, Jm$
3. $\forall x\, Rxx, \exists x\, Rxx$
4. $\exists x\, Px \rightarrow Qc, \exists x(Px \rightarrow Qc)$
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. $\forall x \exists y\, Rxy, \exists x \forall y\, Rxy$
9. $\forall x \exists y\, Rxy, \forall x \exists y\, Ryx$

C. Show that the following sentences are jointly satisfiable:

1. $Ma, \neg Na, Pa, \neg Qa$
2. $Le, e, Leg, \neg Lge, \neg Lgg$
3. $\neg(Ma \wedge \exists x\, Ax), Ma \vee Fa, \forall x(Fx \rightarrow Ax)$
4. $Ma \vee Mb, Ma \rightarrow \forall x \neg Mx$
5. $\forall y\, Gy, \forall x(Gx \rightarrow Hx), \exists y \neg Iy$
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. $\neg \exists x \forall y\, Rxy, \forall x \exists y\, Rxy$

D. 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. $Lab \rightarrow \forall x\, Lxb, \exists x\, Lxb \therefore Lbb$

29 | Reasoning about all interpretations

29.1 Tautologies and contradictions

We can show that a sentence is *not* a tautology just by providing one carefully specified interpretation: an interpretation 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 interpretations in which the sentence is true. A sentence is only a tautology if it is true in *every* interpretation, and there are infinitely many interpretations. We need to reason about all of them, and we cannot do this by dealing with them one by one!

Sometimes, we can reason about all interpretations fairly easily. For example, we can offer a relatively simple argument that '$Raa \lor \neg Raa$' is a tautology:

> Any relevant interpretation will give 'Raa' a truth value. If 'Raa' is true in an interpretation, then '$Raa \lor \neg Raa$' is true in that interpretation. If 'Raa' is false in an interpretation, then $\neg Raa$ is true, and so '$Raa \lor \neg Raa$' is true in that interpretation. These are the only alternatives. So '$Raa \lor \neg Raa$' is true in every interpretation. Therefore, it is a tautology.

This argument is valid, of course, and its conclusion is true. However, it is not an argument in PL. Rather, it is an argument in English *about* PL: it is an argument in the metalanguage.

Note another feature of the argument. Since the sentence in question contained no quantifiers, we did not need to think about how to interpret 'a' and 'R'; the point was just that, however we interpreted them, 'Raa' would have some truth value or other. (We could ultimately have given the same argument concerning SL sentences.)

Here is another bit of reasoning. Consider the sentence '$\forall x(Rxy \lor \neg Ryx)$'. We can show that it is a tautology in PL with the following reasoning.

> Consider some arbitrary interpretation. $\forall x(Rxx \lor \neg Rxx)$ is true in our interpretation iff $Rxx \lor \neg Rxx$ is true, no matter what we take 'x' to name. Consider some arbitrary member of the domain, which, for convenience, we will call Fred—and consider the modified interpretation in which 'x' name Fred. Either Fred bears R to himself or he does not. If Fred bears R to himself, then 'Rxx' will be true on our modified interpretation. And, if 'Rxx is true, then '$Rxx \lor \neg Rcc$' must be true as well. If Fred does

> not bear R to himself, then 'Rxx' is false, so '$\neg Rxx$' will be true on our modified interpretation. And if '$\neg Rxx$' is true, then '$Rxx \vee \neg Rxx$' is true also. So either way, '$Rxx \vee \neg Rxx$' is true. Since there was nothing special about Fred—we might have chosen any object—we see that '$Rxx \vee \neg Rxx$' will be true, *no matter what* 'x' names. So '$\forall x(Rxx \vee \neg Rxx)$' is true in our original interpretation. But we chose that interpretation arbitrarily, so '$\forall x(Rxx \vee \neg Rxx)$' is true in every interpretation. It is therefore a tautology.

This is quite longwinded, but, as things stand, there is no alternative. In order to show that a sentence is a tautology, we must reason about *all* interpretations.

29.2 Other cases

Similar points hold of other cases too. Thus, we must reason about all interpretations if we want to show:

- that a sentence is a contradiction; for this requires that it is false in *every* interpretation.
- that two sentences are equivalent; for this requires that they have the same truth value in *every* interpretation.
- that some sentences are unsatisfiable; for this requires that there is no interpretation in which all of those sentences are true together; i.e. that, in *every* interpretation, at least one of those sentences is false.
- that an argument is valid; for this requires that the conclusion is true in *every* interpretation where the premises are true.
- that some sentences entail another sentence.

The problem is that, with the tools available to you so far, reasoning about all interpretations is a serious challenge! Another example: the following argument is valid:

$$\forall x(Hx \wedge Jx) \quad \therefore \quad \forall x Hx$$

After all, if everything is both H and J, then everything is H. But we can only show thaht this argument is valid by considering what must be true in every interpretation in which the premise is true. To show this, we may reason as follows:

> Consider an arbitrary interpretation in which the premise '$\forall x(Hx \wedge Jx)$' is true. It follows that '$Hx \wedge Jx$' is true, no matter what in the domain we allow 'x' 'to name. If '$Hx \wedge Jx$' is true, then 'Hx' must be true. So, no matter what in the domain we allow 'x' to name, 'Hx' will be true. So it must be that '$\forall x\, Hx$' is true in the interpretation. We've assumed nothing about the interpretation except that it was one in which '$\forall x(Hx \wedge Jx)$' is true, so any interpretation in which '$\forall x(Hx \wedge Jx)$' is true is one in which '$\forall x\, Hx$' is true. The argument is valid!

Even for a simple argument like this one, the reasoning is somewhat complicated.

For one final final example, consider this contradiction:

$$\exists x\, \forall y(Lxy \wedge \neg Lyx)$$

This sentence of PL says that there's something, x, such that, for all y, x bears the relation R to y and y does not bear the relation R back to x. Or, more colloquially, there's something that bears R to everything and is borne R by nothing. It's not obvious that this is a contradiction, but the following clever bit of reasoning will show that it is:

> Suppose that there is an interpretation in which '$\exists x\, \forall y(Lxy \wedge \neg Lyx)$' is true. Then, there must be something in the domain—call it 'Suzy'—such that, if 'x' names Suzy, then the sentence '$\forall y(Lxy \wedge \neg Lyx)$' is true. So consider the modified interpretation in which 'x' names Suzy. Then, '$\forall y(Lxy \wedge \neg Lyx)$' is true in this modified interpretation. If it is true, then '$(Lxy \wedge \neg Lyx)$' must be true, no matter what in the domain 'y' names. So '$(Lxy \wedge \neg Lyx)$' must be true when 'y' names Suzy. So suppose that both 'x' and 'y' 'name Suzy. Then, if '$(Lxy \wedge \neg Lyx)$' is true, 'Lxy' must be true. So Suzy must bear L to herself. And, if '$(Lxy \wedge \neg Lyx)$' is true, then '$\neg Lyx$' must be true. So 'Lyx' must be false. So Suzy must *not* bear L to herself. So Suzy both bears L to herself and doesn't bear L to herself. Contradiction! But we only assumed that there was an interpretation in which '$\exists x\, \forall y(Lxy \wedge \neg Lyx)$' is true. Since this led us to a contradiction, this assumption must be false. So there is no such interpretation. So '$\exists x\, \forall y(Lxy \wedge \neg Lyx)$' is false in *every* interpretation. So it is a contradiction.

The following table summarises whether a single (counter-)interpretation suffices, or whether we must reason about all interpretations.

	Yes	**No**
tautology?	all interpretations	one counter-interpretation
contradiction?	all interpretations	one counter-interpretation
equivalent?	all interpretations	one counter-interpretation
satisfiable?	one interpretation	all interpretations
valid?	all interpretations	one counter-interpretation
entailment?	all interpretations	one counter-interpretation

This might usefully be compared with the table at the end of §13. The key difference resides in the fact that SL concerns truth tables, whereas PL concerns interpretations. This difference is deeply important, since each truth-table only ever has finitely many lines, so that a complete truth table is a relatively tractable object. By contrast, there are infinitely many interpretations for any given sentence(s), so that reasoning about all interpretations can be a deeply tricky business.

PART VII

Natural deduction for PL

30 Basic rules for PL

The language of PL makes use of all of the connectives of SL. So proofs in PL will use all of the basic and derived rules from Part IV. We will also use the proof-theoretic notions (particularly, the symbol '$\vdash$') introduced there. However, we will also need some new basic rules to govern the quantifiers.

30.1 Universal elimination

From the claim that everything is F, you can infer that any particular thing is F. You name it; it's F. So the following should be fine:

$$\begin{array}{r|ll} 1 & \forall x\, Rxxd & \\ \hline 2 & Raad & \forall\text{E } 1 \end{array}$$

We obtained line 2 by dropping the universal quantifier and replacing every instance of 'x' with 'a'. Equally, the following should be allowed:

$$\begin{array}{r|ll} 1 & \forall x\, Rxxd & \\ \hline 2 & Rddd & \forall\text{E } 1 \end{array}$$

We obtained line 2 here by dropping the universal quantifier and replacing every instance of 'x' with 'd'. We could have done the same with any other name we wanted.

This motivates the universal elimination rule (∀E):

$$\begin{array}{r|ll} m & \forall x\, \mathscr{A}(\ldots x \ldots x \ldots) & \\ & \mathscr{A}(\ldots c \ldots c \ldots) & \forall\text{E } m \end{array}$$

A word on the notation: suppose that $\mathscr{A}$ is a formula containing at least one free occurrence of the variable x. We will write this thus:

$$\mathscr{A}(\ldots x \ldots x \ldots)$$

Suppose also that c is a name. Then we will write:

$$\mathscr{A}(\ldots c \ldots c \ldots)$$

for the formula obtained by replacing *every* free occurrence of $\mathscr{x}$ in $\mathscr{A}$ with $\mathscr{c}$. The resulting formula is called a SUBSTITUTION INSTANCE of $\forall\mathscr{x}\mathscr{A}$ and $\exists\mathscr{x}\mathscr{A}$. Also, $\mathscr{c}$ is called the INSTANTIATING NAME. So, for instance:

$$\exists x(Rex \leftrightarrow Fx)$$

is a substitution instance of

$$\forall y\exists x(Ryx \leftrightarrow Fx)$$

with the instantiating name 'e'.

So, the rule $\forall$E says that you can obtain any *substitution instance* of a universally quantified formula by replacing every instance of the quantified variable with any name you like.

I should emphasize that (as with every elimination rule) you can only apply the $\forall$E rule when the universal quantifier is the main operator. So the following is *banned*:

$$\begin{array}{r|ll}
1 & \forall x\, Bx \to Bk & \\
\hline
2 & Bb \to Bk & \text{naughtily attempting to invoke } \forall\text{E } 1
\end{array}$$

This is illegitimate, since '$\forall x$' is not the main operator in line 1. (If you need a reminder as to why this sort of inference should be banned, reread §22.)

It's also important that the rule $\forall$E only allows you to write down a substitution instance of the universally quantified claim. In a substitution instance, you must go through and replace *every* free variable with *the same* name. So, for instance, while the following are all substitution instances of '$\forall x\,(Fxa \lor Gbx)$':

$$Fba \lor Gbb$$
$$Fca \lor Gbc$$
$$Fda \lor Gbd$$

The following are *not* substitution instances of '$\forall x\,(Fxa \lor Gbx)$':

$$Fba \lor Gbx$$
$$Fca \lor Gbd$$

The first sentence is not a substitution instance of '$\forall x\,(Fxa \lor Gbx)$' because we did not replace *every* free occurrence of 'x' in '$(Fxa \lor Gbx)$' with some name. The second sentence is not a substitution instance of '$\forall x\,(Fxa \lor Gbx)$' because we did not replace every free occurrence of 'x' in '$(Fxa \lor Gbx)$' with *the same* name.

Thus, the following use of $\forall$E is illegal:

$$\begin{array}{r|ll}
1 & \forall y\,(Fy \to Gy) & \\
\hline
2 & Fa \to Gb & \text{naughtily attempting to invoke } \forall\text{E } 1
\end{array}$$

since we replaced the first 'y' with 'a', but the second 'y' with 'b'.

30.2 Existential introduction

From the claim that some particular thing is F, you can infer that something is F. So we ought to allow:

$$\begin{array}{r|ll} 1 & Raad & \\ \cline{2-2} 2 & \exists x\, Raax & \exists\text{I } 1 \end{array}$$

Here, we have replaced the name 'd' with a variable 'x', and then existentially quantified over it. Equally, we would have allowed:

$$\begin{array}{r|ll} 1 & Raad & \\ \cline{2-2} 2 & \exists x\, Rxxd & \exists\text{I } 1 \end{array}$$

Here we have replaced both instances of the name 'a' with a variable, and then existentially generalised. But we do not need to replace *both* instances of a name with a variable: if Narcissus loves himself, then there is someone who loves Narcissus. So we also allow:

$$\begin{array}{r|ll} 1 & Raad & \\ \cline{2-2} 2 & \exists x\, Rxad & \exists\text{I } 1 \end{array}$$

Here we have replaced *one* instance of the name 'a' with a variable, and then existentially generalised. These observations motivate our introduction rule, although to explain it, we will need to introduce some new notation.

Where $\mathscr{A}$ is a sentence containing the name $\mathscr{c}$, we can emphasize this by writing '$\mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots)$'. We will write '$\mathscr{A}(\ldots\mathscr{x}\ldots\mathscr{c}\ldots)$' to indicate any formula obtained by replacing *some or all* of the instances of the name $\mathscr{c}$ with the variable $\mathscr{x}$. Armed with this, our introduction rule is:

$$\begin{array}{r|ll} m & \mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots) & \\ & \exists\mathscr{x}\,\mathscr{A}(\ldots\mathscr{x}\ldots\mathscr{c}\ldots) & \exists\text{I } m \end{array}$$

$\mathscr{x}$ must not occur in $\mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots)$

The constraint is included to guarantee that the variable we introduce ends up being bound by the quantifier we introduce. Thus the following is allowed:

$$\begin{array}{r|ll} 1 & Raad & \\ \cline{2-2} 2 & \exists x\, Rxad & \exists\text{I } 1 \\ 3 & \exists y \exists x\, Rxyd & \exists\text{I } 2 \end{array}$$

But this is banned:

1	$Raad$	
2	$\exists x\, Rxad$	∃I 1
3	$\exists x \exists x\, Rxxd$	naughtily attempting to invoke ∃I 2

30.3 Empty domains

The following proof combines our two new rules for quantifiers:

1	$\forall x\, Fx$	
2	Fa	∀E 1
3	$\exists x\, Fx$	∃I 2

Could this be a bad proof? If anything exists at all, then certainly we can infer that something is F, from the fact that everything is F. But what if *nothing* exists at all? Then it is surely vacuously true that everything is F; however, it does not following that something is F, for there is nothing to *be* F. So if we claim that, as a matter of logic alone, '$\exists x\, Fx$' follows from '$\forall x\, Fx$', then we are claiming that, as a matter of *logic alone*, there is something rather than nothing. This might strike us as a bit odd.

Since it is far from clear that logic should tell us that there must be something rather than nothing, we might well be cheating a bit here.

If we refuse to cheat, though, then we pay a high cost. Here are three things that we want to hold on to:

- $\forall x\, Fx \vdash Fa$: after all, that was ∀E.
- $Fa \vdash \exists x\, Fx$: after all, that was ∃I.
- the ability to copy-and-paste proofs together: after all, reasoning works by putting lots of little steps together into rather big chains.

If we get what we want on all three counts, then we have to countenance that $\forall x Fx \vdash \exists x\, Fx$. So, if we get what we want on all three counts, the proof system alone tells us that there is something rather than nothing. And if we refuse to accept that, then we have to surrender one of the three things that we want to hold on to!

Before we start thinking about which to surrender, we might want to ask how *much* of a cheat this is. Granted, it may make it harder to engage in theological debates about why there is something rather than nothing. But the rest of the time, we will get along just fine. So maybe we should just regard our proof system (and PL, more generally) as having a very slightly limited purview. If we ever want to allow for the possibility of *nothing*, then we will have to cast around for a more complicated proof system. But for as long as we are content to ignore that possibility, our proof system is perfectly in order. (As, similarly, is the stipulation that every domain must contain at least one object.)

30.4 Universal introduction

Suppose you had shown of each particular thing that it is F (and that there are no other things to consider). Then you would be justified in claiming that everything is F. This would motivate the following proof rule. If you had established each and every single substitution instance of '$\forall x\, Fx$', then you can infer '$\forall x\, Fx$'.

Unfortunately, that rule would be utterly unusable. To establish each and every single substitution instance would require proving 'Fa', 'Fb', …, 'Fj_2', …, 'Fr_{79002}', …, and so on. Indeed, since there are infinitely many names in PL, this process would never come to an end. So we could never apply that rule. We need to be a bit more cunning in coming up with our rule for introducing universal quantification.

Our cunning thought will be inspired by considering:

$$\forall x\, Fx \mathrel{\therefore} \forall y\, Fy$$

This argument should *obviously* be valid. After all, alphabetical variation ought to be a matter of taste, and of no logical consequence. But how might our proof system reflect this? Suppose we begin a proof thus:

$$\begin{array}{r|ll}
1 & \forall x\, Fx & \\
\hline
2 & Fa & \forall\text{E } 1
\end{array}$$

We have proved 'Fa'. And, of course, nothing stops us from using the same justification to prove 'Fb', 'Fc', …, 'Fj_2', …, 'Fr_{79002}, …, and so on until we run out of space, time, or patience. But reflecting on this, we see that there is a way to prove $F\mathscr{c}$, for any name $\mathscr{c}$. And if we can do it for *any* thing, we should surely be able to say that 'F' is true of *everything*. This therefore justifies us in inferring '$\forall y\, Fy$', thus:

$$\begin{array}{r|ll}
1 & \forall x\, Fx & \\
\hline
2 & Fa & \forall\text{E } 1 \\
3 & \forall y\, Fy & \forall\text{I } 2
\end{array}$$

The crucial thought here is that 'a' was just some *arbitrary* name. There was nothing special about it—we might have chosen any other name—and still the proof would be fine. And this crucial thought motivates the universal introduction rule ($\forall$I):

$$\begin{array}{r|ll}
m & \mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots) & \\
 & \forall\mathscr{x}\,\mathscr{A}(\ldots\mathscr{x}\ldots\mathscr{x}\ldots) & \forall\text{I } m
\end{array}$$

$\mathscr{c}$ must not occur in any undischarged assumption
$\mathscr{x}$ must not occur in $\mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots)$

A crucial aspect of this rule, though, is bound up in the first constraint. This constraint ensures that we are always reasoning at a sufficiently general level. To see the constraint in action, consider this terrible argument:

> Everyone loves Kylie Minogue; therefore everyone loves themselves.

We might symbolize this obviously invalid inference pattern as:

$$\forall x\, Lxk \therefore \forall x\, Lxx$$

Now, suppose we tried to offer a proof that vindicates this argument:

$$\begin{array}{r|ll}
1 & \forall x\, Lxk & \\
\cline{2-2}
2 & Lkk & \forall\text{E } 1 \\
3 & \forall x\, Lxx & \text{naughtily attempting to invoke } \forall\text{I } 2
\end{array}$$

This is not allowed, because 'k' occurred already in an undischarged assumption, namely, on line 1. The crucial point is that, if we have made any assumptions about the object we are working with, then we are not reasoning generally enough to license ∀I.

Although the name may not occur in any *undischarged* assumption, it may occur in a *discharged* assumption. That is, it may occur in a subproof that we have already closed. For example, this is just fine:

$$\begin{array}{r|l|ll}
1 & & Gd & \\
\cline{3-3}
2 & & Gd & \text{R } 1 \\
3 & \multicolumn{2}{l}{Gd \rightarrow Gd} & \rightarrow\text{I } 1\text{–}2 \\
4 & \multicolumn{2}{l}{\forall z(Gz \rightarrow Gz)} & \forall\text{I } 3
\end{array}$$

This tells us that '$\forall z(Gz \rightarrow Gz)$' is a *theorem*. And that is as it should be.

I should emphasise one last point. As per the conventions of §26.3, the use of ∀I requires that we are replacing *every* instance of the name $\mathscr{c}$ in $\mathscr{A}(\ldots \mathscr{x} \ldots \mathscr{x} \ldots)$ with the variable $\mathscr{x}$. If we only replace *some* names and not others, we end up 'proving' silly things. For example, consider the argument:

> Everyone is as old as themselves; so everyone is as old as Judi Dench

We might symbolise this as follows:

$$\forall x\, Oxx \therefore \forall x\, Oxd$$

But now suppose we tried to *vindicate* this terrible argument with the following:

1	$\forall x\, Oxx$	
2	Odd	$\forall$E 1
3	$\forall x\, Oxd$	naughtily attempting to invoke $\forall$I 2

Fortunately, our rules do not allow for us to do this: the attempted proof is banned, since it doesn't replace *every* occurrence of 'd' in line 2 with an 'x'.

30.5 Existential elimination

Suppose we know that *something* is F. The problem is that simply knowing this does not tell us which thing is F. So it would seem that from '$\exists x\, Fx$' we cannot immediately conclude 'Fa', 'Fe_{23}', or any other substitution instance of the sentence. What can we do?

Suppose we know that something is F, and that everything which is F is also G. In (almost) natural English, we might reason thus:

> Since something is F, there is some particular thing which is an F. We do not know anything about it, other than that it's an F, but for convenience, let's call it 'obbie'. So: obbie is F. Since everything which is F is G, it follows that obbie is G. But since obbie is G, it follows that something is G. And nothing depended on which object, exactly, obbie was. So, something is G.

We might try to capture this reasoning pattern in a proof as follows:

1	$\exists x\, Fx$	
2	$\forall x(Fx \to Gx)$	
3	$\quad Fo$	
4	$\quad Fo \to Go$	$\forall$E 2
5	$\quad Go$	$\to$E 4, 3
6	$\quad \exists x\, Gx$	$\exists$I 5
7	$\exists x\, Gx$	$\exists$E 1, 3–6

Breaking this down: we started by writing down our assumptions. At line 3, we made an additional assumption: 'Fo'. This was just a substitution instance of '$\exists x\, Fx$'. On this assumption, we established '$\exists x\, Gx$'. Note that we had made no *special* assumptions about the object named by 'o'; we had *only* assumed that it satisfies 'Fx'. So nothing depends upon which object it is. And line 1 told us that *something* satisfies 'Fx', so our reasoning pattern was perfectly general. We can discharge the specific assumption 'Fo', and simply infer '$\exists x\, Gx$' on its own.

Putting this together, we obtain the existential elimination rule (∃E):

$$\begin{array}{r|lr}
m & \exists x\, \mathscr{A}(\ldots x \ldots x \ldots) & \\
i & \quad\vert\ \mathscr{A}(\ldots c \ldots c \ldots) & \\
 & \quad\vert\ \overline{\phantom{\mathscr{A}(\ldots c \ldots c \ldots)}} & \\
j & \quad\vert\ \mathscr{B} & \\
 & \mathscr{B} & \exists\mathrm{E}\ m,\ i\text{–}j
\end{array}$$

c must not occur in any assumption undischarged before line i
c must not occur in $\exists x\, \mathscr{A}(\ldots x \ldots x \ldots)$
c must not occur in $\mathscr{B}$

As with universal introduction, the constraints are extremely important. To see why, consider the following terrible argument:

> Tim Button is a lecturer. Someone is not a lecturer. So Tim Button is both a lecturer and not a lecturer.

We might symbolize this obviously invalid inference pattern as follows:

$$Lb, \exists x\, \neg Lx \therefore Lb \wedge \neg Lb$$

Now, suppose we tried to offer a proof that vindicates this argument:

$$\begin{array}{r|ll}
1 & Lb & \\
2 & \underline{\exists x\, \neg Lx} & \\
3 & \quad\vert\ \underline{\neg Lb} & \\
4 & \quad\vert\ Lb \wedge \neg Lb & \wedge\mathrm{I}\ 1, 3 \\
5 & Lb \wedge \neg Lb & \text{naughtily attempting to invoke } \exists\mathrm{E}\ \ 2, 3\text{–}4
\end{array}$$

The last line of the proof is not allowed. The name that we used in our substitution instance for '$\exists x\, \neg Lx$' on line 3, namely 'b', occurs in line 4. The this would be no better:

$$\begin{array}{r|ll}
1 & Lb & \\
2 & \underline{\exists x\, \neg Lx} & \\
3 & \quad\vert\ \underline{\neg Lb} & \\
4 & \quad\vert\ Lb \wedge \neg Lb & \wedge\mathrm{I}\ 1, 3 \\
5 & \quad\vert\ \exists x(Lx \wedge \neg Lx) & \exists\mathrm{I}\ 4 \\
6 & \exists x(Lx \wedge \neg Lx) & \text{naughtily attempting to invoke } \exists\mathrm{E}\ \ 2, 3\text{–}5
\end{array}$$

The last line is still not be allowed. For the name that we used in our substitution instance for '$\exists x\, \neg Lx$', namely 'b', occurs in an undischarged assumption, namely line 1.

The moral of the story is this. *If you want to squeeze information out of an existential quantifier, choose a new name for your substitution instance.* That way, you can guarantee that you meet all the constraints on the rule for ∃E.

Practice exercises

A. Explain why these two 'proofs' are *incorrect*. Also, provide interpretations which would invalidate the fallacious argument forms the 'proofs' enshrine:

1	$\forall x\, Rxx$	
2	Raa	∀E 1
3	$\forall y\, Ray$	∀I 2
4	$\forall x \forall y\, Rxy$	∀I 3

1	$\forall x \exists y\, Rxy$	
2	$\exists y\, Ray$	∀E 1
3	$\mid\; Raa$	
4	$\mid\; \exists x\, Rxx$	∃I 3
5	$\exists x\, Rxx$	∃E 2, 3–4

B. The following three proofs are missing their citations (rule and line numbers). Add them, to turn them into bona fide proofs.

1	$\forall x \exists y (Rxy \vee Ryx)$
2	$\forall x\, \neg Rmx$
3	$\exists y (Rmy \vee Rym)$
4	$\mid\; Rma \vee Ram$
5	$\mid\; \neg Rma$
6	$\mid\; Ram$
7	$\mid\; \exists x\, Rxm$
8	$\exists x\, Rx, m$

1	$\forall x (\exists y\, Lxy \rightarrow \forall z\, Lzx)$
2	Lab
3	$\exists y\, Lay \rightarrow \forall z Lza$
4	$\exists y\, Lay$
5	$\forall z\, Lza$
6	Lca
7	$\exists y\, Lcy \rightarrow \forall z\, Lzc$
8	$\exists y\, Lcy$
9	$\forall z\, Lzc$
10	Lcc
11	$\forall x Lxx$

$$\begin{array}{r|l}
1 & \forall x(Jx \to Kx)\\
2 & \exists x\forall y Lxy\\
3 & \underline{\forall x Jx}\\
4 & \quad\vert\ \underline{\forall y Lay}\\
5 & \quad\vert\ Laa\\
6 & \quad\vert\ Ja\\
7 & \quad\vert\ Ja \to Ka\\
8 & \quad\vert\ Ka\\
9 & \quad\vert\ Ka \land Laa\\
10 & \quad\vert\ \exists x(Kx \land Lxx)\\
11 & \exists x(Kx \land Lxx)
\end{array}$$

C. In §22 problem A, we considered fifteen syllogistic figures of Aristotelian logic. Provide proofs for each of the argument forms. NB: You will find it *much* easier if you symbolize (for example) 'No F is G' as '$\forall x(Fx \to \neg Gx)$'.

D. Aristotle and his successors identified other syllogistic forms which depended upon 'existential import'. Symbolize each of these argument forms in PL and offer proofs.

- **Barbari.** Something is H. All G are F. All H are G. So: Some H is F
- **Celaront.** Something is H. No G are F. All H are G. So: Some H is not F
- **Cesaro.** Something is H. No F are G. All H are G. So: Some H is not F.
- **Camestros.** Something is H. All F are G. No H are G. So: Some H is not F.
- **Felapton.** Something is G. No G are F. All G are H. So: Some H is not F.
- **Darapti.** Something is G. All G are F. All G are H. So: Some H is F.
- **Calemos.** Something is H. All F are G. No G are H. So: Some H is not F.
- **Fesapo.** Something is G. No F is G. All G are H. So: Some H is not F.
- **Bamalip.** Something is F. All F are G. All G are H. So: Some H are F.

E. Provide a proof of each claim.

1. $\vdash \forall x Fx \lor \neg\forall x Fx$
2. $\vdash \forall z(Pz \lor \neg Pz)$
3. $\forall x(Ax \to Bx), \exists x Ax \vdash \exists x Bx$
4. $\forall x(Mx \leftrightarrow Nx), Ma \land \exists x Rx, a \vdash \exists x Nx$
5. $\forall x\forall y Gxy \vdash \exists x Gxx$
6. $\vdash \forall x Rxx \to \exists x\exists y Rxy$
7. $\vdash \forall y\exists x(Qy \to Qx)$
8. $Na \to \forall x(Mx \leftrightarrow Ma), Ma, \neg Mb \vdash \neg Na$

9. $\forall x \forall y(Gxy \rightarrow Gyx) \vdash \forall x \forall y(Gxy \leftrightarrow Gyx)$
10. $\forall x(\neg Mx \vee Ljx), \forall x(Bx \rightarrow Ljx), \forall x(Mx \vee Bx) \vdash \forall x Ljx$

F. Write a symbolization key for the following argument, symbolize it, and prove it:

> There is someone who likes everyone who likes everyone that she likes. Therefore, there is someone who likes herself.

G. 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. $\exists x\, Px \rightarrow Qc, \forall x(Px \rightarrow Qc)$

H. For each of the following pairs of sentences: If they are provably equivalent, give proofs to show this. If they are not, construct an interpretation to show that they are not equivalent in PL.

1. $\forall x\, Px \rightarrow Qc, \forall x(Px \rightarrow Qc)$
2. $\forall x \forall y \forall z\, Bxyz, \forall x\, Bxxx$
3. $\forall x \forall y\, Dxy, \forall y \forall x\, Dxy$
4. $\exists x \forall y\, Dxy, \forall y \exists x\, Dxy$
5. $\forall x(Rca \leftrightarrow Rxa), Rca \leftrightarrow \forall x\, Rxa$

I. For each of the following arguments: If it is valid in PL, give a proof. If it is invalid, construct an interpretation to show that it is invalid.

1. $\exists y \forall x\, Rxy \therefore \forall x \exists y\, Rxy$
2. $\forall x \exists y\, Rxy \therefore \exists y \forall x\, Rxy$
3. $\exists x(Px \wedge \neg Qx) \therefore \forall x(Px \rightarrow \neg Qx)$
4. $\forall x(Sx \rightarrow Ta), Sd \therefore Ta$
5. $\forall x(Ax \rightarrow Bx), \forall x(Bx \rightarrow Cx) \therefore \forall x(Ax \rightarrow Cx)$
6. $\exists x(Dx \vee Ex), \forall x(Dx \rightarrow Fx) \therefore \exists x(Dx \wedge Fx)$
7. $\forall x \forall y(Rxy \vee Ryx) \therefore Rjj$
8. $\exists x \exists y(Rxy \vee Ryx) \therefore Rjj$
9. $\forall x\, Px \rightarrow \forall x\, Qx, \exists x \neg Px \therefore \exists x \neg Qx$
10. $\exists x\, Mx \rightarrow \exists x\, Nx, \neg\exists x\, Nx \therefore \forall x \neg Mx$

31 | Constructing proofs for quantifiers

In §16 we discussed strategies for constructing proofs using the basic rules of natural deduction for SL. The same principles apply to the rules for the quantifiers. If we want to prove a quantifier sentence $\forall x \mathscr{A}x$ or $\exists x \mathscr{A}x$ we can work backwards by justifying the sentence we want by $\forall$I or $\exists$I and trying to find a proof of the corresponding premise of that rule. And to use a quantified sentence we apply $\forall$E or $\exists$E, as the case may be.

Specifically, suppose you want to prove $\forall x \mathscr{A}x$. To do so using $\forall$I, we would need a proof of $\mathscr{A}c$ for some name c which does not occur in any undischarged assumption. To apply the corresponding strategy, i.e., to construct a proof of $\forall x \mathscr{A}x$ by working backwards, is thus to write $\mathscr{A}c$ above it and then to continue to try to find a proof of that sentence.

$$\begin{array}{l|ll}
 & \vdots & \\
n & \mathscr{A}c & \\
n+1 & \forall x \mathscr{A}x & \forall\text{I } n
\end{array}$$

$\mathscr{A}c$ is obtained from $\mathscr{A}x$ by replacing every free occurrence of x in $\mathscr{A}x$ by c. For this to work, of course, c must satisfy the special condition. We can ensure that it does by always picking a name that does not already occur in the proof constructed so far. (Of course, if will occur in the proof we end up constructing—just not in an assumption that is undischarged at line $n + 1$.)

To work backward from a sentence $\exists x \mathscr{A}x$ we similarly write a sentence above it that can serve as a justification for an application of the $\exists$I rule, i.e., a sentence of the form $\mathscr{A}c$.

$$\begin{array}{l|ll}
 & \vdots & \\
n & \mathscr{A}c & \\
n+1 & \exists x \mathscr{A}x & \exists\text{I } n
\end{array}$$

This looks just like what we would do if we were working backwards from a universally quantified sentence. The difference is that whereas for $\forall$I we have to pick a name c which does not occur in the proof (so far), for $\exists$I we may and in general must pick a name c which already occurs in the proof. Just like in the case of $\vee$I, it is often not clear which c will work out, and so to avoid having to backtrack you should work backwards from existentially quantified sentences only when all other strategies have been applied.

By contrast, working *forwards* from sentences $\exists x \mathcal{A}x$ generally always works and you won't have to backtrack. Working forwards from an existentially quantified sentence takes into account not just $\exists x \mathcal{A}x$ but also whatever sentence $\mathcal{B}$ you would like to prove. It requires that you set up a subproof above $\mathcal{B}$, wherein $\mathcal{B}$ is the last line, and a substitution instance $\mathcal{A}(c)$ of $\exists x \mathcal{A}x$ is the assumption. In order to ensure that the condition on c that governs $\exists$E, chose a name c which does not already occur in the proof.

$$\begin{array}{l|l l}
 & \vdots & \\
m & \exists x \mathcal{A}x & \\ \cline{2-2}
 & \vdots & \\
n & \quad \mathcal{A}c & \\
 & \quad \vdots & \\
k & \quad \mathcal{B} & \\
k+1 & \mathcal{B} & \exists\text{E } m, n\text{–}k
\end{array}$$

You'll then continue with the goal of proving $\mathcal{B}$, but now inside a subproof in which you have an additional sentence to work with, namely $\mathcal{A}c$.

Lastly, working forwards from $\forall x \mathcal{A}x$ means that you can always write down $\mathcal{A}c$ and justify it using $\forall$E, for any name c. Of course, you wouldn't want to do that willy-nilly. Only certain names c will help in your task of proving whatever goal sentence you are working on. So, like working backward from $\exists x \mathcal{A}x$, you should work forwards from $\forall x \mathcal{A}x$ only after all other strategies have been applied.

Let's consider as an example the argument $\forall x(Ax \to B)$ $\therefore$ $(\exists x\, Ax \to B)$. To start constructing a proof, we write the premise at the top and the conclusion at the bottom.

$$\begin{array}{l|l}
1 & \forall x(Ax \to B) \\ \cline{2-2}
 & \vdots \\
n & \exists x\, Ax \to B
\end{array}$$

The strategies for connectives of SL still apply, and you should apply them in the same order: first work backwards from conditionals, negated sentences, conjunctions, and now also universal quantifiers, then forward from disjunctions and now existential quantifiers, and only then try to apply $\to$E, $\neg$E, $\vee$I, $\forall$E, or $\exists$I. In our case, that means, working backward from the conclusion:

$$
\begin{array}{r|l l}
1 & \forall x(Ax \to B) & \\ \hline
2 & \quad \mid\ \exists x\, Ax & \\
 & \quad \mid\ \vdots & \\
n-1 & \quad \mid\ B & \\
n & \exists x\, Ax \to B & \to\text{I } 2\text{–}(n-1)
\end{array}
$$

Our next step should be to work forward from $\exists x\, Ax$ on line 2. For it we have to pick a name not already in our proof. Since no names appear, we can pick any name, say d

$$
\begin{array}{r|l l}
1 & \forall x(Ax \to B) & \\ \hline
2 & \quad \mid\ \exists x\, Ax & \\
3 & \quad \mid\ \quad \mid\ Ad & \\
 & \quad \mid\ \quad \mid\ \vdots & \\
n-2 & \quad \mid\ \quad \mid\ B & \\
n-1 & \quad \mid\ B & \exists\text{E } 2, 3\text{–}(n-2) \\
n & \exists x\, Ax \to B & \to\text{I } 2\text{–}(n-1)
\end{array}
$$

Now we've exhausted our primary strategies, and it is time to work forward from the premise $\forall x(Ax \to B)$. Applying $\forall$E means we can justify any instance of $Ac \to B$, regardless of what c we choose. Of course, we'll do well to choose d, since that will give us $Ad \to B$ and then we can apply $\to$E to justify B, finishing the proof.

$$
\begin{array}{r|l l}
1 & \forall x(Ax \to B) & \\ \hline
2 & \quad \mid\ \exists x\, Ax & \\
3 & \quad \mid\ \quad \mid\ Ad & \\
4 & \quad \mid\ \quad \mid\ Ad \to B & \forall\text{E } 1 \\
5 & \quad \mid\ \quad \mid\ B & \to\text{E } 4, 3 \\
6 & \quad \mid\ B & \exists\text{E } 2, 3\text{–}5 \\
7 & \exists x\, Ax \to B & \to\text{I } 2\text{–}6
\end{array}
$$

Now let's construct a proof of the converse. We begin with

$$
\begin{array}{r|l}
1 & \exists x\, Ax \to B \\ \hline
 & \vdots \\
n & \forall x(Ax \to B)
\end{array}
$$

Note that the premise is a conditional, not an existentially quantified sentence, so we should not (yet) work forward from it. Working backwards from the conclusion, $\forall x(Ax \rightarrow B)$ leads us to look for a proof of $Ad \rightarrow B$:

1	$\exists x\, Ax \rightarrow B$	
	$\vdots$	
$n-1$	$Ad \rightarrow B$	
n	$\forall x(Ax \rightarrow B)$	$\forall$I $n-1$

And working backward from $Ad \rightarrow B$ means we should set up a subproof with Ad as an assumption and B as the last line:

1	$\exists x\, Ax \rightarrow B$	
2	$\vert\; A(d)$	
	$\vert\; \vdots$	
$n-2$	$\vert\; B$	
$n-1$	$Ad \rightarrow B$	$\rightarrow$I 2–$(n-2)$
n	$\forall x(Ax \rightarrow B)$	$\forall$I $n-1$

Now we can work forwards from the premise on line 1. That's a conditional, and its consequent happens to be the sentence B we are trying to justify. So we should look for a proof of its antecedent, $\exists x\, Ax$. Of course, that is now readily available, by $\exists$I from line 2, and we're done:

1	$\exists x\, Ax \rightarrow B$	
2	$\vert\; Ad$	
3	$\vert\; \exists x\, Ax$	$\exists$I 2
4	$\vert\; B$	$\rightarrow$E 1, 3
5	$Ad \rightarrow B$	$\rightarrow$I 2–4
6	$\forall x(Ax \rightarrow B)$	$\forall$I 5

32 Conversion of quantifiers

In this section, we will add some additional rules to the basic rules of the previous section. These govern the interaction of quantifiers and negation.

In §21, we noted that $\neg\exists x\mathscr{A}$ is logically equivalent to $\forall x\neg\mathscr{A}$. We will add some rules to our proof system that govern this. In particular, we add:

$$\begin{array}{r|ll} m & \forall x\neg\mathscr{A} & \\ & \neg\exists x\mathscr{A} & \text{CQ } m \end{array}$$

and

$$\begin{array}{r|ll} m & \neg\exists x\mathscr{A} & \\ & \forall x\neg\mathscr{A} & \text{CQ } m \end{array}$$

Equally, we add:

$$\begin{array}{r|ll} m & \exists x\neg\mathscr{A} & \\ & \neg\forall x\mathscr{A} & \text{CQ } m \end{array}$$

and

$$\begin{array}{r|ll} m & \neg\forall x\mathscr{A} & \\ & \exists x\neg\mathscr{A} & \text{CQ } m \end{array}$$

Practice exercises

A. Show in each case that the sentences are provably inconsistent:

1. $Sa \rightarrow Tm, Tm \rightarrow Sa, Tm \land \neg Sa$

2. $\neg\exists x\, Rxa, \forall x\forall y\, Ryx$
3. $\neg\exists x\exists y\, Lxy, Laa$
4. $\forall x(Px \to Qx), \forall z(Pz \to Rz), \forall y\, Py, \neg Qa \land \neg Rb$

B. Show that each pair of sentences is provably equivalent:

1. $\forall x(Ax \to \neg Bx), \neg\exists x(Ax \land Bx)$
2. $\forall x(\neg Ax \to Bd), \forall x\, Ax \lor Bd$

C. In §22, we considered what happens when we move quantifiers 'across' various logical operators. Show that each pair of sentences is provably equivalent:

1. $\forall x(Fx \land Ga), \forall x\, Fx \land Ga$
2. $\exists x(Fx \lor Ga), \exists x\, Fx \lor Ga$
3. $\forall x(Ga \to Fx), Ga \to \forall x\, Fx$
4. $\forall x(Fx \to Ga), \exists x\, Fx \to Ga$
5. $\exists x(Ga \to Fx), Ga \to \exists x\, Fx$
6. $\exists x(Fx \to Ga), \forall x\, Fx \to Ga$

NB: the variable 'x' does not occur in 'Ga'. When all the quantifiers occur at the beginning of a sentence, that sentence is said to be in *prenex normal form*. These equivalences are sometimes called *prenexing rules*, since they give us a means for putting any sentence into prenex normal form.

33 | Derived rules

As in the case of SL, we first introduced some rules for PL as basic (in §30), and then added some further rules for conversion of quantifiers (in §32). In fact, the CQ rules should be regarded as *derived* rules, for they can be derived from the *basic* rules of §30. (The point here is as in §19.) Here is a justification for the first CQ rule:

```
1 │ ∀x¬Ax
  ├──────
2 │ │ ∃xAx
  │ ├──────
3 │ │ │ Ac
  │ │ ├─────
4 │ │ │ ¬Ac     ∀E 1
5 │ │ │ ⊥       ⊥I 4, 3
6 │ │ ⊥         ∃E 2, 3–5
7 │ ¬∃xAx       ¬I 2–6
```

Here is a justification of the third CQ rule:

```
1 │ ∃x¬Ax
  ├──────
2 │ │ ∀xAx
  │ ├──────
3 │ │ │ ¬Ac
  │ │ ├──────
4 │ │ │ Ac      ∀E 2
5 │ │ │ ⊥       ⊥I 3, 4
6 │ │ ⊥         ∃E 1, 3–5
7 │ ¬∀xAx       ¬I 2–6
```

This explains why the CQ rules can be treated as derived. Similar justifications can be offered for the other two CQ rules.

Practice exercises

A. Offer proofs which justify the addition of the second and fourth CQ rules as derived rules.

34 Proof-theoretic and semantic concepts

We have used two different turnstiles in this book. This:

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vdash \mathscr{C}$$

means that there is some proof which starts with assumptions $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ and ends with $\mathscr{C}$ (and no undischarged assumptions other than $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$). This is a *proof-theoretic notion*.

By contrast, this:

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vDash \mathscr{C}$$

means that no valuation (or interpretation) makes all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ true and $\mathscr{C}$ false. This concerns assignments of truth and falsity to sentences. It is a *semantic notion*.

It cannot be emphasized enough that these are different notions. But we can emphasize it a bit more: *They are different notions.*

Once you have internalised this point, continue reading.

Although our semantic and proof-theoretic notions are different, there is a deep connection between them. To explain this connection,we will start by considering the relationship between tautologies and theorems.

To show that a sentence is a theorem, you need only produce a proof. Granted, 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 about all possible interpretations. 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, to show that a sentence is *not* a theorem is hard. We would need to reason about all (possible) proofs. That is very difficult. However, to show that a sentence is not a tautology, you need only construct an interpretation in which the sentence is false. Granted, it may be hard to come up with the interpretation; but once you have done so, it is relatively straightforward to check what truth value it assigns to a sentence. 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*. As a result, if we provide a proof of $\mathscr{A}$ on no assumptions, and thus show that $\mathscr{A}$ is a theorem, i.e. $\vdash \mathscr{A}$, we can legitimately infer that $\mathscr{A}$ is a tautology, i.e., $\vDash \mathscr{A}$. Similarly, if we construct an interpretation in which $\mathscr{A}$ is false and thus show that it is not a tautology, i.e. $\nvDash \mathscr{A}$, it follows that $\mathscr{A}$ is not a theorem, i.e. $\nvdash \mathscr{A}$.

More generally, we have the following powerful result:

$$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vdash \mathscr{B} \textbf{ iff } \mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \vDash \mathscr{B}$$

This shows that, whilst provability and entailment are *different* notions, they are extensionally equivalent. As such:

- An argument is *valid* iff *the conclusion can be proved from the premises*.
- Two sentences are *logically equivalent* iff they are *provably equivalent*.
- Sentences are *satisfiable* iff they are *not provably inconsistent*.

For this reason, you can pick and choose when to think in terms of proofs and when to think in terms of valuations/interpretations, doing whichever is easier for a given task. The table on the next page summarises which is (usually) easier.

It is intuitive that provability and semantic entailment should agree. But—let us repeat this—do not be fooled by the similarity of the symbols '$\vDash$' and '$\vdash$'. These two symbols have very different meanings. The fact that provability and semantic entailment agree is not an easy result to come by.

In fact, demonstrating that provability and semantic entailment agree is, very decisively, the point at which introductory logic becomes intermediate logic.

	Yes	**No**
Is $\mathscr{A}$ a **tautology**?	give a proof which shows $\vdash \mathscr{A}$	give an interpretation in which $\mathscr{A}$ is false
Is $\mathscr{A}$ a **contradiction**?	give a proof which shows $\vdash \neg\mathscr{A}$	give an interpretation in which $\mathscr{A}$ is true
Are $\mathscr{A}$ and $\mathscr{B}$ **equivalent**?	give two proofs, one for $\mathscr{A} \vdash \mathscr{B}$ and one for $\mathscr{B} \vdash \mathscr{A}$	give an interpretation in which $\mathscr{A}$ and $\mathscr{B}$ have different truth values
Are $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ **satisfiable**?	give an interpretation in which all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are true	prove a contradiction from assumptions $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$
Is $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \therefore \mathscr{C}$ **valid**?	give a proof with assumptions $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ and concluding with $\mathscr{C}$	give an interpretation in which each of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ is true and $\mathscr{C}$ is false

Appendices

A | Symbolic notation

1.1 Alternative nomenclature

Sentence logic. SL goes by other names. Sometimes it is called *truth-functional logic*, to emphasize the fact that it deals only with assignments of truth and falsity to sentences, and that its connectives are all truth-functional. Sometimes it is called *propositional logic*, on the idea that it deals fundamentally with propositions.

Predicate logic. PL goes by other names. Sometimes it is called *first-order logic*, in contrast to *second-order* logic, which allows you to use quantify over predicates as well as names. That is, in second-order logic, you can write '$\forall X \exists Y (Xa \leftrightarrow Ya)$'. Sometimes PL is called *quantified logic*, because it makes use of quantifiers.

Sentences. Some texts call sentences *well-formed formulas*. Since 'well-formed formula' is such a long and cumbersome phrase, they then abbreviate this as *wff*. This is both barbarous and unnecessary (such texts do not countenance 'ill-formed formulas'). We have stuck with 'sentence'.

Valuations. Some texts call valuations *truth-assignments*, or *truth-value assignments*.

Expressive adequacy. Some texts describe TFL as *truth-functionally complete*, rather than expressively adequate.

***n*-place predicates.** We have chosen to call predicates 'one-place', 'two-place', 'three-place', etc. Other texts respectively call them 'monadic', 'dyadic', 'triadic', etc. Still other texts call them 'unary', 'binary', 'ternary', etc.

Names. In PL, we have used 'a', 'b', 'c', for names. Some texts call these 'constants'. Other texts do not mark any difference between names and variables in the syntax. Those texts focus simply on whether the symbol occurs *bound* or *unbound*.

Domains. Some texts describe a domain as a 'domain of discourse', or a 'universe of discourse'.

1.2 Alternative symbols

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. This appendix presents some common symbols, so that you can recognize them if you encounter them in an article or in another book.

Negation. Two commonly used symbols are the *hoe*, '$\neg$', and the *swung dash* or *tilda*, '$\sim$'. 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 '$\sim$'. Older texts sometimes indicate negation by a line over the formula being negated, e.g., $\overline{A \wedge B}$. Some texts use '$x \neq y$' to abbreviate '$\neg x = y$'.

Disjunction. The symbol '$\vee$' is typically used to symbolize inclusive disjunction. One etymology is from the Latin word 'vel', meaning 'or'.

Conjunction. Conjunction is often symbolized with the *ampersand*, '&'. The ampersand is a decorative form of the Latin word 'et', which means 'and'. This symbol is commonly used in natural English writing (e.g. 'Smith & Sons'), and so even though it is a natural choice, many logicians use a different symbol to avoid confusion between the object and metalanguage: as a symbol in a formal system, the ampersand is not the English word '&'. The most common choice now is '$\wedge$', which 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 a few common symbols for the material conditional: the *arrow*, '$\rightarrow$', and the *hook*, '$\supset$'. Some texts use '$\Rightarrow$'.

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. Those that use '$\Rightarrow$' for the material conditional use '$\Leftrightarrow$' for the biconditional.

Quantifiers. The universal quantifier is typically symbolized as a rotated 'A', and the existential quantifier as a rotated, 'E'. 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, they might write '$(x)Px$' where we would write '$\forall x\, Px$'.

These alternative typographies are summarised below:

negation	$\neg\mathscr{A}$, $\sim\mathscr{A}$, $\overline{\mathscr{A}}$
conjunction	$\mathscr{A} \wedge \mathscr{B}$, $\mathscr{A} \,\&\, \mathscr{B}$, $\mathscr{A} \bullet \mathscr{B}$, $\mathscr{A}\mathscr{B}$
disjunction	$\mathscr{A} \vee \mathscr{B}$
conditional	$\mathscr{A} \rightarrow \mathscr{B}$, $\mathscr{A} \supset \mathscr{B}$, $\mathscr{A} \Rightarrow \mathscr{B}$
biconditional	$\mathscr{A} \leftrightarrow \mathscr{B}$, $\mathscr{A} \equiv \mathscr{B}$, $\mathscr{A} \Leftrightarrow \mathscr{B}$
universal quantifier	$\forall x\, \mathscr{A}$, $(x)\mathscr{A}$

B | Alternative proof systems

In formulating our natural deduction system, we treated certain rules of natural deduction as *basic*, and others as *derived*. However, we could equally well have taken various different rules as basic or derived. We will illustrate this point by considering some alternative treatments of negation, disjunction, and the quantifiers.

2.1 Alternative negation rules

Some systems take the following rule as their basic negation introduction rule:

$$
\begin{array}{l|l|ll}
m & & \mathscr{A} & \\ \cline{3-3}
n-1 & & \mathscr{B} & \\
n & & \neg\mathscr{B} & \\
 & \multicolumn{2}{l}{\neg\mathscr{A}} & \neg\text{I}^{*}\ m\text{–}n
\end{array}
$$

and a corresponding version of ¬E as their basic negation elimination rule:

$$
\begin{array}{l|l|ll}
m & & \neg\mathscr{A} & \\ \cline{3-3}
n-1 & & \mathscr{B} & \\
n & & \neg\mathscr{B} & \\
 & \multicolumn{2}{l}{\mathscr{A}} & \neg\text{E}^{*}\ m\text{–}n
\end{array}
$$

Using these two rules, we could we could have avoided all use of the symbol '⊥' altogether.[1] The resulting system would have had fewer rules than ours.

Another way to deal with negation is to use either LEM or DNE as a basic rule and introduce ¬E as a derived rule.

[1] Again, P.D. Magnus's original version of this book went the other way.

2.2 Alternative disjunction elimination

Some systems take DS as their basic rule for disjunction elimination. Such systems can then treat the ∨E rule as a derived rule. For they might offer the following proof scheme:

```
m       | $\mathscr{A} \vee \mathscr{B}$
i       | | $\mathscr{A}$
        | |----
j       | | $\mathscr{C}$
k       | | $\mathscr{B}$
        | |----
l       | | $\mathscr{C}$
n       | $\mathscr{A} \rightarrow \mathscr{C}$        $\rightarrow$I $i$–$j$
n+1     | $\mathscr{B} \rightarrow \mathscr{C}$        $\rightarrow$I $k$–$l$
n+2     | | $\neg\mathscr{C}$
        | |----
n+3     | | | $\mathscr{A}$
        | | |----
n+4     | | | $\mathscr{C}$                          $\rightarrow$E $n+3$, $n$
n+5     | | | $\bot$                                 $\bot$I $n+2$, $n+4$
n+6     | | $\neg\mathscr{A}$                        $\neg$I $n+3$–$n+5$
n+7     | | $\mathscr{B}$                            DS $m$, $n+6$
n+8     | | $\mathscr{C}$                            $\rightarrow$E $n+7$, $n+1$
n+9     | | $\bot$                                   $\bot$I $n+2$, $n+8$
n+10    | $\mathscr{C}$                              $\neg$E $n+2$–$n+9$
```

So why did we choose to take ∨E as basic, rather than DS?[2] Our reasoning is that DS involves the use of '¬' in the statement of the rule. It is in some sense 'cleaner' for our disjunction elimination rule to avoid mentioning *other* connectives.

2.3 Alternative quantification rules

An alternative approach to the quantifiers is to take as basic the rules for ∀I and ∀E from §30, and also two CQ rule which allow us to move from $\forall x \neg \mathscr{A}$ to $\neg \exists x \mathscr{A}$ and vice versa.[3]

[2] P.D. Magnus's original version of this book went the other way.

[3] Warren Goldfarb follows this line in *Deductive Logic*, 2003, Hackett Publishing Co.

Taking only these rules as basic, we could have derived the ∃I and ∃E rules provided in §30. To derive the ∃I rule is fairly simple. Suppose $\mathscr{A}$ contains the name c, and contains no instances of the variable x, and that we want to do the following:

$$\begin{array}{r|l}
m & \mathscr{A}(\ldots c \ldots c \ldots) \\
k & \exists x \mathscr{A}(\ldots x \ldots c \ldots)
\end{array}$$

This is not yet permitted, since in this new system, we do not have the ∃I rule. We can, however, offer the following:

$$\begin{array}{r|l|ll}
m & \multicolumn{2}{l}{\mathscr{A}(\ldots c \ldots c \ldots)} & \\ \hline
m+1 & & \neg\exists x \mathscr{A}(\ldots x \ldots c \ldots) & \\ \cline{3-3}
m+2 & & \forall x \neg\mathscr{A}(\ldots x \ldots c \ldots) & \text{CQ } m+1 \\
m+3 & & \neg\mathscr{A}(\ldots c \ldots c \ldots) & \forall\text{E } m+2 \\
m+4 & & \bot & \bot\text{I } m+3, m \\
m+5 & \multicolumn{2}{l}{\exists x \mathscr{A}(\ldots x \ldots c \ldots)} & \neg\text{E } m+1\text{–}m+4
\end{array}$$

To derive the ∃E rule is rather more subtle. This is because the ∃E rule has an important constraint (as, indeed, does the ∀I rule), and we need to make sure that we are respecting it. So, suppose we are in a situation where we *want* to do the following:

$$\begin{array}{r|l|l}
m & \multicolumn{2}{l}{\exists x \mathscr{A}(\ldots x \ldots x \ldots)} \\
i & & \mathscr{A}(\ldots c \ldots c \ldots) \\ \cline{3-3}
j & & \mathscr{B} \\
k & \multicolumn{2}{l}{\mathscr{B}}
\end{array}$$

where c does not occur in any undischarged assumptions, or in $\mathscr{B}$, or in $\exists x \mathscr{A}(\ldots x \ldots x \ldots)$. Ordinarily, we would be allowed to use the ∃E rule; but we are not here assuming that we have access to this rule as a basic rule. Nevertheless, we could offer the following, more complicated derivation:

$$\begin{array}{l|lll}
m & \exists x \mathscr{A}(\ldots x \ldots x \ldots) & & \\
i & & \vert\ \underline{\mathscr{A}(\ldots c \ldots c \ldots)} & \\
j & & \vert\ \mathscr{B} & \\
k & \mathscr{A}(\ldots c \ldots c \ldots) \to \mathscr{B} & & \to\text{I } i\text{–}j \\
k+1 & & \vert\ \underline{\neg\mathscr{B}} & \\
k+2 & & \vert\ \neg\mathscr{A}(\ldots c \ldots c \ldots) & \text{MT } k, k+1 \\
k+3 & & \vert\ \forall x \neg\mathscr{A}(\ldots x \ldots x \ldots) & \forall\text{I } k+2 \\
k+4 & & \vert\ \neg\exists x \mathscr{A}(\ldots x \ldots x \ldots) & \text{CQ } k+3 \\
k+5 & & \vert\ \bot & \bot\text{I } k+4, m \\
k+6 & \mathscr{B} & & \neg\text{E } k+1\text{–}k+5
\end{array}$$

We are permitted to use $\forall$I on line $k+3$ because c does not occur in any undischarged assumptions or in $\mathscr{B}$. The entries on lines $k+4$ and $k+1$ contradict each other, because c does not occur in $\exists x \mathscr{A}(\ldots x \ldots x \ldots)$.

Armed with these derived rules, we could now go on to derive the two remaining CQ rules, exactly as in §33.

So, why did we start with all of the quantifier rules as basic, and then derive the CQ rules?

Our first reason is that it seems more intuitive to treat the quantifiers as on a par with one another, giving them their own basic rules for introduction and elimination.

Our second reason relates to the discussion of alternative negation rules. In the derivations of the rules of $\exists$I and $\exists$E that we have offered in this section, we invoked $\neg$E. But, as we mentioned earlier, $\neg$E is a contentious rule. So, if we want to move to a system which abandons $\neg$E, but which still allows us to use existential quantifiers, we will want to take the introduction and elimination rules for the quantifiers as basic, and take the CQ rules as derived. (Indeed, in a system without $\neg$E, LEM, and DNE, we will be *unable* to derive the CQ rule which moves from $\neg\forall x \mathscr{A}$ to $\exists x \neg\mathscr{A}$.)

C | Quick reference

3.1 Characteristic Truth Tables

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

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{A} \wedge \mathscr{B}$	$\mathscr{A} \vee \mathscr{B}$	$\mathscr{A} \rightarrow \mathscr{B}$	$\mathscr{A} \leftrightarrow \mathscr{B}$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

3.2 Symbolization

Sentential Connectives

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	$(P \leftrightarrow Q)$
P unless Q	$(P \vee Q)$

Predicates

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 $\neg\exists x(Fx \wedge Gx)$

3.3 Basic deduction rules for SL

Conjunction

$$\begin{array}{r|ll}
m & \mathscr{A} & \\
n & \mathscr{B} & \\
 & \mathscr{A} \wedge \mathscr{B} & \wedge\text{I } m, n
\end{array}$$

$$\begin{array}{r|ll}
m & \mathscr{A} \wedge \mathscr{B} & \\
 & \mathscr{A} & \wedge\text{E } m
\end{array}$$

$$\begin{array}{r|ll}
m & \mathscr{A} \wedge \mathscr{B} & \\
 & \mathscr{B} & \wedge\text{E } m
\end{array}$$

Conditional

$$\begin{array}{r|ll}
i & \quad\begin{array}{|l} \mathscr{A} \\ \hline \end{array} & \\
j & \quad\begin{array}{|l} \mathscr{B} \end{array} & \\
 & \mathscr{A} \rightarrow \mathscr{B} & \rightarrow\text{I } i\text{–}j
\end{array}$$

$$\begin{array}{r|ll}
m & \mathscr{A} \rightarrow \mathscr{B} & \\
n & \mathscr{A} & \\
 & \mathscr{B} & \rightarrow\text{E } m, n
\end{array}$$

Negation

$$\begin{array}{r|ll}
i & \quad\begin{array}{|l} \mathscr{A} \\ \hline \end{array} & \\
j & \quad\begin{array}{|l} \bot \end{array} & \\
 & \neg\mathscr{A} & \neg\text{I } i\text{–}j
\end{array}$$

$$\begin{array}{r|ll}
i & \quad\begin{array}{|l} \neg\mathscr{A} \\ \hline \end{array} & \\
j & \quad\begin{array}{|l} \bot \end{array} & \\
 & \mathscr{A} & \neg\text{E } i\text{–}j
\end{array}$$

Contradiction

$$\begin{array}{r|ll}
m & \neg\mathscr{A} & \\
n & \mathscr{A} & \\
 & \bot & \bot\text{I } m, n
\end{array}$$

$$\begin{array}{r|ll}
m & \bot & \\
 & \mathscr{A} & \bot\text{E } m
\end{array}$$

Disjunction

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

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

$$\begin{array}{r|ll}
m & \mathscr{A} \vee \mathscr{B} & \\
i & \quad\begin{array}{|l} \mathscr{A} \\ \hline \end{array} & \\
j & \quad\begin{array}{|l} \mathscr{C} \end{array} & \\
k & \quad\begin{array}{|l} \mathscr{B} \\ \hline \end{array} & \\
l & \quad\begin{array}{|l} \mathscr{C} \end{array} & \\
 & \mathscr{C} & \vee\text{E } m, i\text{–}j, k\text{–}l
\end{array}$$

Biconditional

```
i │ │ 𝒜
  │ ├──
j │ │ ℬ
k │ │ ℬ
  │ ├──
l │ │ 𝒜
  │ 𝒜 ↔ ℬ      ↔I i–j, k–l
```

```
m │ 𝒜 ↔ ℬ
n │ 𝒜
  │ ℬ          ↔E m, n
```

```
m │ 𝒜 ↔ ℬ
n │ ℬ
  │ 𝒜          ↔E m, n
```

3.4 Derived rules for SL

Disjunctive syllogism

$$\begin{array}{r|ll} m & \mathscr{A} \vee \mathscr{B} & \\ n & \neg\mathscr{A} & \\ & \mathscr{B} & \text{DS } m, n \end{array}$$

$$\begin{array}{r|ll} m & \mathscr{A} \vee \mathscr{B} & \\ n & \neg\mathscr{B} & \\ & \mathscr{A} & \text{DS } m, n \end{array}$$

Excluded middle

$$\begin{array}{r|l|ll} i & & \underline{\mathscr{A}} & \\ j & & \mathscr{B} & \\ k & & \underline{\neg\mathscr{A}} & \\ l & & \mathscr{B} & \\ & \mathscr{B} & & \text{LEM } i\text{–}j, k\text{–}l \end{array}$$

Reiteration

$$\begin{array}{r|ll} m & \mathscr{A} & \\ & \mathscr{A} & \text{R } m \end{array}$$

Modus Tollens

$$\begin{array}{r|ll} m & \mathscr{A} \rightarrow \mathscr{B} & \\ n & \neg\mathscr{B} & \\ & \neg\mathscr{A} & \text{MT } m, n \end{array}$$

Double-negation elimination

$$\begin{array}{r|ll} m & \neg\neg\mathscr{A} & \\ & \mathscr{A} & \text{DNE } m \end{array}$$

De Morgan Rules

$$\begin{array}{r|ll} m & \neg(\mathscr{A} \vee \mathscr{B}) & \\ & \neg\mathscr{A} \wedge \neg\mathscr{B} & \text{DeM } m \end{array}$$

$$\begin{array}{r|ll} m & \neg\mathscr{A} \wedge \neg\mathscr{B} & \\ & \neg(\mathscr{A} \vee \mathscr{B}) & \text{DeM } m \end{array}$$

$$\begin{array}{r|ll} m & \neg(\mathscr{A} \wedge \mathscr{B}) & \\ & \neg\mathscr{A} \vee \neg\mathscr{B} & \text{DeM } m \end{array}$$

$$\begin{array}{r|ll} m & \neg\mathscr{A} \vee \neg\mathscr{B} & \\ & \neg(\mathscr{A} \wedge \mathscr{B}) & \text{DeM } m \end{array}$$

3.5 Basic deduction rules for PL

Universal elimination

$$\begin{array}{r|ll} m & \forall x\mathscr{A}(\ldots x\ldots x\ldots) & \\ & \mathscr{A}(\ldots c\ldots c\ldots) & \forall\text{E } m \end{array}$$

Existential introduction

$$\begin{array}{r|ll} m & \mathscr{A}(\ldots c\ldots c\ldots) & \\ & \exists x\mathscr{A}(\ldots x\ldots c\ldots) & \exists\text{I } m \end{array}$$

x must not occur in $\mathscr{A}(\ldots c\ldots c\ldots)$

Universal introduction

$$\begin{array}{r|ll} m & \mathscr{A}(\ldots c\ldots c\ldots) & \\ & \forall x\mathscr{A}(\ldots x\ldots x\ldots) & \forall\text{I } m \end{array}$$

c must not occur in any undischarged assumption

x must not occur in $\mathscr{A}(\ldots c\ldots c\ldots)$

Existential elimination

$$\begin{array}{r|ll} m & \exists x\mathscr{A}(\ldots x\ldots x\ldots) & \\ i & \quad \mid\ \underline{\mathscr{A}(\ldots c\ldots c\ldots)} & \\ j & \quad \mid\ \mathscr{B} & \\ & \mathscr{B} & \exists\text{E } m, i\text{–}j \end{array}$$

c must not occur in any undischarged assumption, in $\exists x\mathscr{A}(\ldots x\ldots x\ldots)$, or in $\mathscr{B}$

3.6 Derived rules for PL

$$\begin{array}{r|ll} m & \forall x\neg\mathscr{A} & \\ & \neg\exists x\mathscr{A} & \text{CQ } m \end{array}$$

$$\begin{array}{r|ll} m & \exists x\neg\mathscr{A} & \\ & \neg\forall x\mathscr{A} & \text{CQ } m \end{array}$$

$$\begin{array}{r|ll} m & \neg\exists x\mathscr{A} & \\ & \forall x\neg\mathscr{A} & \text{CQ } m \end{array}$$

$$\begin{array}{r|ll} m & \neg\forall x\mathscr{A} & \\ & \exists x\neg\mathscr{A} & \text{CQ } m \end{array}$$

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.