forall$x$

Bristol 2017-18

Last updated November 8, 2017

By P.D. Magnus and Tim Button

With additions by J. Robert Loftis
Remixed and revised Aaron Thomas-Bolduc and Richard Zach
With alterations by Catrin Campbell-Moore

This book is based on `forall`x: *Calgary* by

Aaron Thomas-Bolduc & Richard Zach
University of Calgary

used under a CC BY-SA 4.0 license, which is based on `forall`x: *Cambridge*, by

Tim Button
University of Cambridge

used under a CC BY-SA 3.0 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-SA 3.0 license.

then altered by

Catrin Campbell-Moore
University of Bristol

It includes additional material from `forall`x by P.D. Magnus and *Metatheory* by Tim Button, both used under a CC BY-SA 3.0 license, and from `forall`x: *Lorain County Remix*, by Cathal Woods and J. Robert Loftis, used under a CC BY-SA 4.0 license.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 license. You are free to copy and redistribute the material in any medium or format, and remix, transform, and build upon the material for any purpose, even commercially, under the following terms:

- ▷ You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- ▷ If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

The LaTeX source for this book is available on GitHub.

Contents

Preface v

1 Key notions of logic 1

1 Arguments 2
2 Valid arguments 5
3 Other logical notions 9

2 Truth-functional logic 14

4 First steps to symbolization 15
5 Connectives 18
6 Sentences of TFL 29
7 Use and mention 34

3 Truth tables 38

8 Characteristic truth tables 39
9 Truth-functional connectives 41
10 Complete truth tables 44
11 Semantic concepts 49
12 Truth table shortcuts 56
13 Partial truth tables 60

4 Natural deduction for TFL 65

14 The very idea of natural deduction 66
15 Basic rules for TFL 68
16 Additional rules for TFL 85
17 Proof-theoretic concepts 89
18 Proof strategies 92
19 Derived rules 93
20 Soundness and completeness 97

5 First-order logic 103

21 Building blocks of FOL 104
22 Sentences with one quantifier 112
23 Multiple quantifiers 121
24 Sentences of FOL 129

6 Interpretations 132

25 Extensionality 133
26 Truth in FOL 137
27 Semantic concepts 142
28 Using interpretations 143
29 Reasoning about all interpretations 148

7 Natural deduction for FOL 151

30 Basic rules for FOL 152
31 Conversion of quantifiers 162
32 Derived rules 164
33 Proof-theoretic and semantic concepts 165

8 Identity 168

34 Identity 169
35 Definite descriptions 173
36 Semantics for FOL with identity 179
37 Rules for identity 181

9 Advanced Topics (Non-examinable) 184

38 Normal forms and expressive completeness 185

Appendices 193

A Symbolic notation 193
B Alternative proof systems 196
C Quick reference 201

Glossary 207

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 is 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 consistent, 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 eight parts. The first part deals introduces the topic and notions of logic in an informal way, without introducing a formal language yet. Parts II–IV concern truth-functional languages. In it, sentences are formed from basic sentences using a number of connectives ('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). Parts V–VII deal with a more complicated language, that of first-order logic. It includes, in addition to the connectives

of truth-functional logic, also names, predicates, identity, and the so-called quantifiers. These additional elements of the language make it much more expressive that the truth-functional language, 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 first-order logic are defined semantically, using interpretations, and proof-theoretically, using a more complex version of the formal derivation system introduced in Part IV. Part VIII covers an advanced topic: that of expressive adequacy of the truth-functional connectives.

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. The central terms are listed in a glossary at the very end.

This book is based on a text originally written by P. D. Magnus and revised and expanded by Tim Button and independently by J. Robert Loftis. Aaron Thomas-Bolduc and Richard Zach have combined elements of these texts into the present version, changed some of the terminology and examples, and added material of their own. Catrin Campbell-Moore has then made very minor alterations for the Bristol course. The resulting text is licensed under a Creative Commons Attribution-ShareAlike 4.0 license.

Chapter 1

Key notions of logic

Arguments 1

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

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 guideline, the following words are often used to indicate an argument's conclusion:

so, therefore, hence, thus, accordingly, consequently

Similarly, 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 problems that review and explore the material covered in the chapter. There is no substitute for actually working through some problems, because logic is more about a way of thinking than it is about memorizing facts.

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.

Valid arguments 2

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 as follows. 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.

To determine whether or not the premises of an argument are true is often a very important matter. 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.

The crucial thing about a valid argument is that it is impossible for the premises to be true whilst 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 were 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. The argument is therefore 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.

Sometimes arguments that seem plausible will be invalid. Consider:

If I have freshers flu, then I have a sore throat.
I have a sore throat.
∴ I have freshers flu.

This is invalid because it is possible that I have a sore throat without having freshers flu; I might have just been shouting too loudly yesterday. This sort of fallacy is called 'Affirming the Consequent'.

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 (deductively) 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 (deductively) valid arguments from the invalid ones.

Practice exercises

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

1. Socrates is a man.
2. All men are carrots.

∴ Socrates is a carrot.

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

∴ Abe Lincoln was born in Illinois.

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

∴ Abe Lincoln will not die.

1. Abe Lincoln was either from France or from Luxemborg.
2. Abe Lincoln was not from Luxemborg.

∴ Abe Lincoln was from France.

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

∴ The world will not end today.

1. Joe is now 19 years old.
2. 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.

Other logical notions 3

In §2, we introduced the idea of a valid argument. We will want to introduce some more ideas that are important in logic.

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 inconsistent with each other, they cannot all be true at the same time. This motivates the following definition:

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

Conversely, B1 and B2 are *jointly impossible*.

We can ask about the 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 contingency

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 is 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. A jointly impossible set 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 not jointly possible
8. A necessary truth that is also a necessary falsehood
9. A jointly possible collection of sentences that are all necessary falsehoods

Chapter 2

Truth-functional logic

First steps to symbolization 4

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 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 this chapter, 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 *truth-functional logic*, or TFL.

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

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

We will use atomic sentence 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 atomic sentence of TFL, 'A', as symbolizing the English sentence 'It is raining outside', and the atomic sentence of TFL, '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 an atomic sentence of TFL. From the point of view of TFL, an atomic sentence is just a letter. It can be used to build more complex sentences, but it cannot be taken apart.

Connectives 5

In the previous chapter, we considered symbolizing fairly basic English sentences with atomic sentences of TFL. This leaves us wanting to deal with the English expressions 'and', 'or', 'not', and so forth. These are *connectives*—they can be used to form new sentences out of old ones. In TFL, we will make use of logical connectives to build complex sentences from atomic components. There are five logical connectives in TFL. This table summarises 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	'... or ...'
$\rightarrow$	conditional	'If ... then ...'
$\leftrightarrow$	biconditional	'... if and only if ...'

These are not the only connectives of English of interest. Others are, e.g., 'unless', 'neither ... nor ...', and 'because'. We will see that the first two can be expressed by the connectives 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 an atomic sentence. We might offer this symbolization key:

B: Mary is in Barcelona.

Since sentence 2 is obviously related to the 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 TFL 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 TFL-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 atomic sentence of TFL.

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 atomic sentences of TFL 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 connective 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 TFL.

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 an atomic sentence of TFL. 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 TFL 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 an atomic sentence, we need to replace 'she' with 'Barbara'. So we can paraphrase sentence 14 as, '*Both* Barbara is energetic, *and* Barbara is not athletic'. The second conjunct contains a negation, so we paraphrase further: '*Both* Barbara is energetic *and it is not the case that* Barbara is athletic'. Now we can symbolize this with the TFL 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'. TFL does not (and cannot) preserve these nuances.

Sentence 15 raises similar issues. There is a contrastive structure, but this is not something that TFL 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 TFL 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 brackets 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. Brackets help us to keep track of things like the *scope* of the negation.

5.3 Disjunction

Consider these sentences:

18. Fatima will play videogames, or she will watch movies.
19. 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 connective 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 '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 '..., 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 TFL 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 TFL, 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 TFL symbol '$\vee$' always symbolizes *inclusive or*, we can symbolize an *exclusive or* in TFL. 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 connective 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 connective '$\rightarrow$' 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 '$M \to D$'.

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 '$((D \to M) \wedge (M \to D))$'. 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 TFL symbol to deal with *exclusive or*, we do not really need a new TFL 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 TFL 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 connectives of TFL. We can use them together to symbolize many kinds of sentences. An especially difficult case is when we use the English-language connective '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 equivalent. They are also equivalent to the following:

> If you do not wear a jacket, then you will catch a cold.
> If you do not catch a cold, then you must have worn a jacket.
> Either you will wear a jacket or you will catch a cold.

And we know how to symbolise these sentences. We will use the symbolization key:

> J: You will wear a jacket.

D: You will catch a cold.

and can then give the symbolizations '$\neg J \to D$', '$\neg D \to J$' and '$J \vee D$'.

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

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

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 TFL.

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 TFL.

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 TFL.

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 TFL.

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 TFL.

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 TFL.

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 TFL.

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 TFL. 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 connectives? Is there any way to symbolize an *exclusive or* using only one connective?

Sentences of TFL 6

The sentence 'either apples are red, or berries are blue' is a sentence of English, and the sentence '$(A \vee B)$' is a sentence of TFL. Although we can identify sentences of English when we encounter them, we do not have a formal definition of 'sentence of English'. But in this chapter, we will offer a complete *definition* of what counts as a sentence of TFL. This is one respect in which a formal language like TFL is more precise than a natural language like English.

6.1 Expressions

We have seen that there are three kinds of symbols in TFL:

Atomic sentences	$A, B, C, \ldots, Z$
with subscripts, as needed	$A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$
Connectives	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
Brackets	(,)

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

6.2 Sentences

Of course, many expressions of TFL will be total gibberish. For example

$$\wedge)A\neg$$

is an expression of TFL but not a sentence. We want to know when an expression of TFL amounts to a *sentence*.

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

Instead, we will describe the process by which sentences can be *constructed*. Consider negation: Given any sentence $\mathscr{A}$ of TFL, $\neg\mathscr{A}$ is a sentence of TFL. (Why the funny fonts? We return to this in §7.)

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

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. 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 TFL. 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 atomic sentences, 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 TFL. 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 an atomic sentence of TFL, so we know that 'D' is a sentence by the first clause of the definition. So for a compound sentence like '$\neg\neg\neg D$', we must apply the definition repeatedly. Eventually we arrive at the atomic sentences from which the sentence is built up.

Next, consider the example '$\neg(P \wedge \neg(\neg Q \vee R))$'. Looking at the second clause of the definition, 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 an atomic sentence, 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, and both are!

Ultimately, every sentence is constructed nicely out of atomic sentences. When we are dealing with a *sentence* other than an atomic sentence, we can see that there must be some sentential connective that was introduced *last*, when constructing the sentence. We call that connective the MAIN LOGICAL OPERATOR of the sentence. In the case of '$\neg\neg\neg D$', the main logical operator is the very first '$\neg$' sign. In the case of '$(P \wedge \neg(\neg Q \vee R))$', the main logical operator is '$\wedge$'. In the case of '$((\neg E \vee F) \rightarrow \neg\neg G)$', the main logical operator is '$\rightarrow$'.

To keep track of the structure of an expression we can draw it out in a tree, which we will sometimes call a FORMATION TREE. For example for '$\neg(P \wedge \neg(\neg Q \vee R))$' the tree would be:

$$
\begin{array}{c}
\neg(P \wedge \neg(\neg Q \vee R)) \\
| \\
(P \wedge \neg(\neg Q \vee R)) \\
/ \quad \backslash \\
P \quad \neg(\neg Q \vee R) \\
/ \quad \backslash \\
\neg Q \quad R \\
| \\
Q
\end{array}
$$

This displays clearly the recursive structure of the tree. The sentences further up the tree are formed by one of the formation rules from the sentences further down the tree. Ideally we would also label the splittings with their main connectives, but unfortunately I haven't yet worked out how to do that in LaTeX.

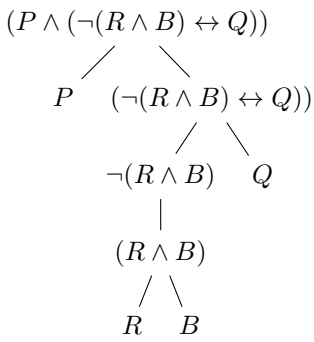
The recursive structure of sentences in TFL will be important when we consider the circumstances under which a particular sentence would be true or false. The sentence '$\neg\neg\neg D$' is true if and only if the sentence '$\neg\neg D$' is false, and so on through the structure of the sentence, until we arrive at the atomic components. We will return to this point in Part 3.

The recursive structure of sentences in TFL 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. One can use formation trees to work this out: the subsentence for which '$\neg$' is the main logical operator will be the lowest sentence in which the connective appears.

Consider a sentence like:

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

This has a formation tree:



The scope of '$\neg$' is '$\neg(R \wedge B)$'. It is the subsentence in which '$\neg$' is the main logical operator. We have worked that out by drawing the formation tree and finding the lowest sentence in which '$\neg$' appears.

More generally:

> The SCOPE of a connective (in a sentence) is the subsentence for which that connective is the main logical operator.

6.3 Bracketing conventions

Strictly speaking, the brackets 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 brackets 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 TFL, 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* brackets 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 brackets 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 brackets. To make things a bit easier on the eyes, we will allow ourselves to use square brackets, '[' 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

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

rather more clearly as follows:

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

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

Practice exercises

A. For each of the following: (a) Is it a sentence of TFL, strictly speaking? (b) Is it a sentence of TFL, allowing for our relaxed bracketing conventions?

1. (A)
2. $J_{374} \lor \neg J_{374}$
3. $\neg\neg\neg\neg F$
4. $\neg \land S$
5. $(G \land \neg G)$
6. $(A \to (A \land \neg F)) \lor (D \leftrightarrow E)$
7. $[(Z \leftrightarrow S) \to W] \land [J \lor X]$
8. $(F \leftrightarrow \neg D \to J) \lor (C \land D)$

B. Are there any sentences of TFL that contain no atomic sentences? Explain your answer.

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

$$\big[(H \to I) \lor (I \to H)\big] \land (J \lor K)$$

Use and mention 7

In this chapter, 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, TFL, and so we are often *mentioning* expressions from TFL.

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: TFL. 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 italic uppercase letters for atomic sentences of TFL:

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

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

- D is an atomic sentence of TFL.

Obviously, we are trying to come out with an English sentence that says something about the object language (TFL), but 'D' is a sentence of TFL, 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 an atomic sentence of TFL.

The general point is that, whenever we want to talk in English about some specific expression of TFL, 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 TFL. We also want to be able to talk about *any arbitrary* sentence of TFL. Indeed, we had to do this in §6, when we presented the recursive definition of a sentence of TFL. We used uppercase script letters to do this, namely:

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

These symbols do not belong to TFL. Rather, they are part of our (augmented) metalanguage that we use to talk about *any* expression of TFL. To repeat the second clause of the recursive definition of a sentence of TFL, 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 TFL expression. 'A' is a particular atomic sentence of TFL.

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 TFL sentence, since '$\mathscr{A}$' is a symbol of (augmented) English rather than a symbol of TFL.

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. So then 2′ is the correct form of the recursive clauses.

7.4 Quotation conventions for arguments

One of our main purposes for using TFL is to study arguments, and that will be our concern in Parts 3 and 4. 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 TFL. Thus we might ask whether the argument whose premises are the TFL sentences 'A' and '$A \to C$', and whose conclusion is the TFL 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 TFL.)

Chapter 3

Truth tables

Characteristic truth tables 8

Any non-atomic sentence of TFL is composed of atomic sentences with sentential connectives. The truth value of the compound sentence depends only on the truth value of the atomic sentences that comprise it. In order to know the truth value of '$(D \wedge E)$', for instance, you only need to know the truth value of 'D' and the truth value of 'E'.

We introduced five connectives 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 $\mathscr{A}$ is true 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 TFL. 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: $\mathscr{A} \to \mathscr{B}$ 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.

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

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.

Truth-functional connectives 9

9.1 The idea of truth-functionality

Let's introduce an important idea.

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

Every connective in TFL 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 TFL sentence, we only need to know the truth value of its components.

This is what gives TFL its name: it is *truth-functional logic*.

In plenty of languages there are connectives 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 a connective of English, but it is not *truth-functional*.

9.2 Symbolizing versus translating

All of the connectives of TFL 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 TFL, 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

TFL. As remarked in §5, TFL 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 TFL sentence, perhaps '$L \wedge N$'.

We keep saying that we use TFL sentences to *symbolize* English sentences. Many other textbooks talk about *translating* English sentences into TFL. However, a good translation should preserve certain facets of meaning, and—as we have just pointed out—TFL 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 TFL sentence 'L' should *mean* that Dana is a logician, and that the TFL sentence 'N' should *mean* that Dana is a nice person, but TFL just is totally unequipped to deal with *meaning*. The preceding symbolization key is doing no more and no less than stipulating that the TFL sentence 'L' should take the same truth value as the English sentence 'Dana is a logician' (whatever that might be), and that the TFL 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 TFL sentence as *symbolizing* an English sentence, we are stipulating that the TFL 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 TFL 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.

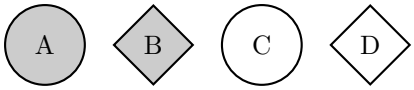
As it happens, Lara has drawn the following:

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

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)

Now, recall that every connective of TFL 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 TFL 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 TFL'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 TFL, but that many English conditionals cannot be represented adequately using '→'. TFL is an intrinsically limited language.

Complete truth tables 10

So far, we have considered assigning truth values to TFL sentences indirectly. We have said, for example, that a TFL 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 atomic sentences of TFL.

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 atomic sentence '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 atomic sentences 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

To know how to proceed we will want to look at the structure of the sentence, which we can do by producing a formation tree as in §6.2:

$$
\begin{array}{ccc}
 & (H \wedge I) \to H & \\
 & / \quad \backslash & \\
(H \wedge I) & & H \\
/ \quad \backslash & & \\
H \quad I & &
\end{array}
$$

To fill in the truth table we will be working up the tree. We have to have assigned truth values to all components lower in the tree to work out the node connecting them. So we should start with the lowest sentence that we have not yet got a truth value for: '$(H \wedge I)$'. This is a conjunction, $(\mathscr{A} \wedge \mathscr{B})$, 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 $(\mathscr{A} \wedge \mathscr{B})$, 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.

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

		$\mathscr{A}$	$\to$	$\mathscr{B}$
H	I	$(H \wedge I)$	$\to$	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 '$(H \wedge I) \to H$' 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 '$(H \wedge I) \to H$' 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	$(H \wedge I) \rightarrow H$
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 atomic sentences. 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 atomic sentences in the table. A sentence that contains only one atomic sentence 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 atomic sentences requires four lines for a complete truth table, as in the characteristic truth tables for our binary connectives, and as in the complete truth table for '$(H \wedge I) \rightarrow H$'.

A sentence that contains three atomic sentences 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 atomic sentences 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 atomic sentences, then it must have 2^n lines.

In order to fill in the columns of a complete truth table, begin with the rightmost atomic sentence 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 atomic sentence, 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 atomic sentences 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 brackets

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 TFL cares about (see §9) – which of the two sentences we assert (or deny). Even though the order of the brackets 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 brackets. *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 brackets.

Practice exercises

A. Draw formation trees and offer complete truth tables for each of the following:

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

Semantic concepts 11

In the previous section, we introduced the idea of a valuation and showed how to determine the truth value of any TFL 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 Entailment and validity

Logic is particularly useful for evaluating arguments, in particular to help us see if an argument is *valid*: logic can help us see if an argument is valid in virtue of its form.

In §2 we introduce the notion of validity: an argument is valid if it's impossible for all the premises to be true and the conclusion false. We investigated validity as it applies to English language arguments, but we also have TFL arguments. The logical substitute notion of validity replaces 'impossible' by '*logically* impossible', where logical impossibility is characterised by there being no valuations with the property of interest.

> The TFL argument '$\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n \therefore \mathscr{B}$' is VALID iff there are no valuations where the premises are true and the conclusion false.

We will be using validity a lot in this course, and we have a corresponding notion that makes the claim that an argument is valid: that is the notion of *entailment*.

> The TFL sentences $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ ENTAIL the sentence $\mathscr{C}$ iff there are no valuations where all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ true and $\mathscr{C}$ false.

We can see that entailment and validity are sister-notions:

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

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

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

In short, we have a way to test for the validity of English arguments. First, we symbolize them in TFL, 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.

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 TFL 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 TFL. Rather, it is a symbol of our metalanguage, augmented English (recall the difference between object language and metalanguage from §7). So the metalanguage sentence:

- $P, P \to Q \vDash Q$

is just an abbreviation for the English sentence:

- The TFL sentences 'P' and '$P \to Q$' entail 'Q'

Note that there is no limit on the number of TFL 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.

'$\vDash$' versus '$\to$'

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

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

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

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

> '$\to$' is a sentential connective of TFL.
> '$\vDash$' is a symbol of augmented English.

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

11.2 Tautologies and contradictions

We have similar substitutes for the notions we introduced in §3. There we explained *necessary truth* and *necessary falsity*. Both notions have surrogates in TFL. We will start with a surrogate for necessary truth.

> $\mathscr{A}$ is a TAUTOLOGY iff it is true on every valuation.

We can determine whether a sentence is a tautology 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, '$(H \wedge I) \to H$' is a tautology.

Our main example of a tautology is $A \vee \neg A$. Whatever the weather is like you know it's either raining or not. Similarly here, whether or not A is true, we already know $A \vee \neg A$. We can check this with a truth table.

A	$A \vee \neg A$
T	T **T** F T
F	F **T** T F

Since there is a T under the main connective on every line of the truth table, $A \vee \neg A$ is true, whatever the valuation is, i.e. whether A is true or false.

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

Our core example of a contradiction is $A \wedge \neg A$. Whether A is true or false, $A \wedge \neg A$ is false. This can again be checked using truth tables, observing that there is an F under the main connective in each line of the truth table.

11.3 Logical equivalence

Here is a similar useful notion:

> $\mathscr{A}$ and $\mathscr{B}$ are LOGICALLY EQUIVALENT 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 logically equivalent. Again, it is easy to test for logical equivalence using truth tables. Consider the sentences '$\neg(P \vee Q)$' and '$\neg P \wedge \neg Q$'. Are they logically 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 logically equivalent.

11.4 Consistency

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 JOINTLY LOGICALLY CONSISTENT iff there is some valuation which makes them all true.

Derivatively, sentences are jointly logically inconsistent if there is no valuation that makes them all true. Again, it is easy to test for joint logical consistency 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 TFL, we would have to use two different atomic sentences – 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 TFL, 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 TFL, we would offer something like '$\neg(G \to M)$'. Now, '$\neg(G \to M)$' entails 'G' (again, check this with a truth table). So if we symbolize sentence 3 in TFL, 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 '$\neg(G \to M)$' 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 '$G \to \neg M$'. Now, if atheists are right, and there is no God, then 'G' is false and so '$G \to \neg M$' is true, and the puzzle disappears. However, if 'G' is false, '$G \to M$', 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 TFL) that can *only* handle truth-functional connectives. 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 TFL, before we can adequately discuss its limits, and consider alternatives.

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 jointly consistent, or jointly inconsistent:

1. $A \to A$, $\neg A \to \neg A$, $A \wedge A$, $A \vee A$
2. $A \vee B$, $A \to C$, $B \to C$
3. $B \wedge (C \vee A)$, $A \to B$, $\neg(B \vee C)$

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. $A \rightarrow (A \wedge \neg A) \therefore \neg A$
3. $A \vee (B \rightarrow A) \therefore \neg A \rightarrow \neg B$
4. $A \vee B, B \vee C, \neg A \therefore B \wedge C$
5. $(B \wedge A) \rightarrow C, (C \wedge A) \rightarrow B \therefore (C \wedge B) \rightarrow A$

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

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. $[(A \wedge B) \leftrightarrow B] \rightarrow (A \rightarrow B)$

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. $A \wedge \neg A$ and $\neg B \leftrightarrow B$
3. $[(A \vee B) \vee C]$ and $[A \vee (B \vee C)]$
4. $A \vee (B \wedge C)$ and $(A \vee B) \wedge (A \vee C)$
5. $[A \wedge (A \vee B)] \rightarrow B$ and $A \rightarrow B$

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

G. Determine whether each collection of sentences is jointly consistent or jointly inconsistent using a complete truth table.

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

H. Determine whether each collection of sentences is jointly consistent or jointly inconsistent, using a complete truth table.

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

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

1. $A \vee \big[A \to (A \leftrightarrow A)\big] \therefore A$
2. $A \vee B$, $B \vee C$, $\neg B \therefore A \wedge C$
3. $A \to B$, $\neg A \therefore \neg B$
4. A, $B \therefore \neg(A \to \neg B)$
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 $(\mathscr{A} \wedge \mathscr{B}) \to \mathscr{C}$ 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 jointly inconsistent. 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.

Truth table shortcuts 12

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 atomic sentence, but can simply refer back to them. So you can speed things up by writing:

P	Q	$(P \vee Q) \leftrightarrow \neg P$
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	$(\neg P \vee \neg Q) \vee \neg P$
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	$\neg (P \wedge \neg Q) \wedge \neg P$
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:

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

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	$\neg L \to (J \vee L)$	$\neg L$	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	$\neg L \to (J \vee L)$	$\neg L$	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. $(B \wedge D) \leftrightarrow [A \leftrightarrow (A \vee C)]$

Partial truth tables 13

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 '$(U \wedge T) \rightarrow (S \wedge W)$' is *not* a tautology. We set up a PARTIAL TRUTH TABLE:

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

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

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

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

S	T	U	W	$(U \wedge T) \rightarrow (S \wedge W)$
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 atomic sentences.

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.

Consistency. To show that some sentences are jointly consistent, 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 jointly inconsistent, 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 truth table	one-line partial truth table
contradiction?	complete truth table	one-line partial truth table
equivalent?	complete truth table	one-line partial truth table
consistent?	one-line partial truth table	complete truth table
valid?	complete truth table	one-line partial truth table
entailment?	complete truth table	one-line partial truth table

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 jointly consistent, or jointly inconsistent:

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

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 $A \to B$
4. $(A \to B)$ and $(\neg B \to \neg A)$
5. $\neg(A \wedge B)$ and $\neg A \vee \neg B$
6. $((U \to (X \vee X)) \vee U)$ and $\neg(X \wedge (X \wedge U))$
7. $((C \wedge (N \leftrightarrow C)) \leftrightarrow C)$ and $(\neg\neg\neg N \to C)$
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 jointly consistent or jointly inconsistent. Justify your answer with a complete or partial truth table where appropriate.

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

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

1. $A \to (A \wedge \neg A) \therefore \neg A$
2. $A \vee B$, $A \to B$, $B \to A \therefore A \leftrightarrow B$
3. $A \vee (B \to A) \therefore \neg A \to \neg B$
4. $A \vee B$, $A \to B$, $B \to A \therefore A \wedge B$
5. $(B \wedge A) \to C$, $(C \wedge A) \to B \therefore (C \wedge B) \to A$
6. $\neg(\neg A \vee \neg B)$, $A \to \neg C \therefore A \to (B \to C)$
7. $A \wedge (B \to C)$, $\neg C \wedge (\neg B \to \neg A) \therefore C \wedge \neg C$

8. $A \wedge B$, $\neg A \to \neg C$, $B \to \neg D \therefore A \vee B$
9. $A \to B \therefore (A \wedge B) \vee (\neg A \wedge \neg B)$
10. $\neg A \to B$,$\neg B \to C$,$\neg C \to A \therefore \neg A \to (\neg B \vee \neg C)$

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. $A \to C$, $E \to (D \vee B)$, $B \to \neg D \therefore (A \vee C) \vee (B \to (E \wedge D))$
4. $A \vee B$, $C \to A$, $C \to B \therefore A \to (B \to C)$
5. $A \to B$, $\neg B \vee A \therefore A \leftrightarrow B$

Chapter 4

Natural deduction for TFL

The very idea of natural deduction 14

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 TFL, this led us to develop truth tables. Each line of a complete truth table corresponds to a valuation. So, when faced with a TFL 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 TFL:

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

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:

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

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 TFL 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 TFL), we will try to select a few basic rules of inference. Some of these will govern the behaviour of the sentential connectives. 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 (both in 1934). However, the natural deduction system that we will consider is based largely around work by Frederic Fitch (first published in 1952).

Basic rules for TFL 15

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

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

We will start a proof by writing the premise:

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

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

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

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

Similarly, suppose we wanted to consider:

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

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

1	$A \vee B$
2	$\neg(A \wedge C)$
3	$\neg(B \wedge \neg D)$

and we are hoping to conclude with some line:

n	$\neg C \vee D$

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 connectives.

15.2 Reiteration

The very first rule is so breathtakingly obvious that it is surprising we bother with it at all.

If you already have shown something in the course of a proof, the *reiteration rule* allows you to repeat it on a new line. For example:

4	$A \wedge B$	
$\vdots$	$\vdots$	
10	$A \wedge B$	R 4

This indicates that we have written '$A \wedge B$' on line 4. Now, at some later line—line 10, for example—we have decided that we want to repeat this. So we write it down again. We also add a citation which justifies what we have written. In this case, we write 'R', to indicate that we are using the reiteration rule, and we write '4', to indicate that we have applied it to line 4.

Here is a general expression of the rule:

m	$\mathscr{A}$	
	$\mathscr{A}$	R m

The point is that, if any sentence $\mathscr{A}$ occurs on some line, then we can repeat $\mathscr{A}$ on later lines. Each line of our proof must be justified by some rule, and here we have 'R m'. This means: Reiteration, applied to line m.

Two things need emphasising. First '$\mathscr{A}$' is not a sentence of TFL. Rather, it a symbol in the metalanguage, which we use when we want to talk about any sentence of TFL (see §7). Second, and similarly, 'm' is not a numeral that will appear on a proof. Rather, it is a symbol in the metalanguage, which we use when we want to talk about any line number of a 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 underscore the point that the rule may be applied at any point.

15.3 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:

8	R	
15	L	
	$R \wedge L$	$\wedge$I 8, 15

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

8	R	
15	L	
	$L \wedge R$	$\wedge$I 15, 8

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

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

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 TFL. Rather, they are symbols in the metalanguage, which we use when we want to talk about any sentence of TFL (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):

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

and equally:

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

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. (One point, might be worth emphasising: you can only apply this rule when conjunction is the main logical 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:

$$\begin{array}{ll} & [(A \vee B) \rightarrow (C \vee D)] \wedge \neg(E \vee F) \\ \therefore & \neg(E \vee F) \wedge [(A \vee B) \rightarrow (C \vee D)] \end{array}$$

The main logical 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:

1	$[(A \vee B) \rightarrow (C \vee D)] \wedge \neg(E \vee F)$

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

1	$[(A \vee B) \rightarrow (C \vee D)] \wedge \neg(E \vee F)$	
2	$[(A \vee B) \rightarrow (C \vee D)]$	$\wedge$E 1
3	$\neg(E \vee F)$	$\wedge$E 1

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

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

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 a staggering 256 lines; our formal proof required only four lines.

It is worth giving another example. Way 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:

1	$A \wedge (B \wedge C)$

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

1	$A \wedge (B \wedge C)$	
2	A	$\wedge$E 1
3	$B \wedge C$	$\wedge$E 1
4	B	$\wedge$E 3
5	C	$\wedge$E 3

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

1	$A \wedge (B \wedge C)$	
2	A	$\wedge$E 1
3	$B \wedge C$	$\wedge$E 1
4	B	$\wedge$E 3
5	C	$\wedge$E 3
6	$A \wedge B$	$\wedge$I 2, 4
7	$(A \wedge B) \wedge C$	$\wedge$I 6, 5

Recall that our official definition of sentences in TFL 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 requirement that it is applied to two different sentences. So we can formally prove 'A' from 'A' as follows:

1	A	
2	$A \wedge A$	∧I 1, 1
3	A	∧E 2

Simple, but effective. In fact this shows that we didn't need to have the rule of Reiteration as we could always argue by ∧I then ∧E. But for ease we will allow Reiteration as a basic rule so you don't have to argue this way.

15.4 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):

m	$\mathscr{A} \rightarrow \mathscr{B}$	
n	$\mathscr{A}$	
	$\mathscr{B}$	→E m, n

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	| $R \wedge L$	∧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 '$L \to (R \wedge L)$':

1	R	
2	| L	
3	| $R \wedge L$	∧I 1, 2
4	$L \to (R \wedge L)$	→I 2–3

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:

m	| $\mathscr{A}$	
	| ⋮	
n	| $\mathscr{B}$	
	$\mathscr{A} \to \mathscr{B}$	→I m–n

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 $\rightarrow$I in action. Suppose we want to consider the following:

$$P \rightarrow Q, Q \rightarrow R \therefore P \rightarrow R$$

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

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

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

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

15.5 Additional assumptions and subproofs

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

Consider this proof:

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

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	$\quad B$	
3	$\quad B$	R 2
4	$B \to B$	$\to$I 2–3
5	B	naughty attempt to invoke $\to$E 4, 3

If we were allowed to do this, it would be a disaster. It would allow us to prove any atomic sentence letter from any other atomic 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:

> Any rule whose citation requires mentioning individual lines can mention any earlier lines, *except* for those lines which occur within a closed subproof.

This stipulation rules out the disastrous attempted proof above. The rule of $\to$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	$\quad B$	
3	$\quad\quad C$	
4	$\quad\quad A \wedge B$	$\wedge$I 1, 2
5	$\quad C \to (A \wedge B)$	$\to$I 3–4
6	$B \to (C \to (A \wedge B))$	$\to$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	$\quad B$	
3	$\quad\quad C$	
4	$\quad\quad A \wedge B$	$\wedge$I 1, 2
5	$\quad C \to (A \wedge B)$	$\to$I 3–4
6	$B \to (C \to (A \wedge B))$	$\to$I 2–5
7	$C \to (A \wedge B)$	naughty attempt to invoke $\to$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:

> Any rule whose citation requires mentioning an entire subproof can mention any earlier subproof, *except* for those subproofs which 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.

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 $\rightarrow$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.

15.6 Biconditional

Will not be required in exam.

The rules for the biconditional will just encode the fact that we didn't need to introduce the symbol and could have always written '$(\mathscr{A} \rightarrow \mathscr{B}) \wedge (\mathscr{B} \rightarrow \mathscr{A})$' instead of $\mathscr{A} \leftrightarrow \mathscr{B}$. We thus introduce the rules allowing you to move between these formulations:

m	$\mathscr{A} \leftrightarrow \mathscr{B}$	
	$(\mathscr{A} \rightarrow \mathscr{B}) \wedge (\mathscr{B} \rightarrow \mathscr{A})$	$\leftrightarrow$E m

m	$(\mathscr{A} \rightarrow \mathscr{B}) \wedge (\mathscr{B} \rightarrow \mathscr{A})$	
	$\mathscr{A} \leftrightarrow \mathscr{B}$	$\leftrightarrow$I m

15.7 Disjunction

Suppose Ludwig is reactionary. Then Ludwig is reactionary or libertarian. After all, to say that Ludwig is 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 reactionary *or* he is a kumquat. Equally, it follows that Ludwig is reactionary *or* that kumquats are the only fruit. Equally, it follows that 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):

m	$\mathscr{A}$	
	$\mathscr{A} \vee \mathscr{B}$	$\vee$I m

and

m	$\mathscr{A}$	
	$\mathscr{B} \vee \mathscr{A}$	$\vee$I m

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

1	M	
2	$M \vee ([(A \leftrightarrow B) \rightarrow (C \wedge D)] \leftrightarrow [E \wedge F])$	$\vee$I 1

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

The disjunction elimination rule is, though, slightly trickier. Suppose that 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 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 ($\vee$E) rule:

m	$\mathscr{A} \vee \mathscr{B}$	
i	$\quad \mathscr{A}$	
	$\quad \vdots$	
j	$\quad \mathscr{C}$	
k	$\quad \mathscr{B}$	
	$\quad \vdots$	
l	$\quad \mathscr{C}$	
	$\mathscr{C}$	$\vee$E m, i–j, k–l

This is obviously a bit clunkier to write down than our previous rules, but the point is fairly simple. Suppose we have some disjunction, $\mathscr{A} \vee \mathscr{B}$. 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:

$$(P \wedge Q) \vee (P \wedge R) \therefore P$$

An example proof might run thus:

1	$(P \wedge Q) \vee (P \wedge R)$	
2	$\quad P \wedge Q$	
3	$\quad P$	$\wedge$E 2
4	$\quad P \wedge R$	
5	$\quad P$	$\wedge$E 4
6	P	$\vee$E 1, 2–3, 4–5

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	$\quad B$	
5	$\quad A \wedge B$	$\wedge$I 2, 4
6	$\quad (A \wedge B) \vee (A \wedge C)$	$\vee$I 5
7	$\quad C$	
8	$\quad A \wedge C$	$\wedge$I 2, 7
9	$\quad (A \wedge B) \vee (A \wedge C)$	$\vee$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 will come 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.

15.8 Law of Excluded Middle

We actually add another rule for how to introduce a disjunction. There are special kinds of disjunctions that don't need further justification: sentences like $A \vee \neg A$. In §11.2 we saw that $A \vee \neg A$ is a tautology: it is true on all valuations. The rule *Law of Excluded Middle* encodes this fact: it simply says that you are always allowed to write $\mathscr{A} \vee \neg \mathscr{A}$:

	$\mathscr{A} \lor \neg\mathscr{A}$	LEM

As always, $\mathscr{A}$ can be whatever you want, e.g. $(A \land (B \to C))$. Then the rule tells us, e.g. that you can write $(A \land (B \to C)) \lor \neg(A \land (B \to C))$ on any line of the proof.

To see the rule in action, consider:

$$P \therefore (P \land D) \lor (P \land \neg D)$$

Here is a proof corresponding with the argument:

1	P	
2	$D \lor \neg D$	LEM
3	$\quad D$	
4	$\quad P \land D$	$\land$I 1, 3
5	$\quad (P \land D) \lor (P \land \neg D)$	$\lor$I 4
6	$\quad \neg D$	
7	$\quad P \land \neg D$	$\land$I 1, 6
8	$\quad (P \land D) \lor (P \land \neg D)$	$\lor$I 7
9	$(P \land D) \lor (P \land \neg D)$	$\lor$E 2, 3–5, 6–8

15.9 Contradiction

We have only one connective left to deal with: negation, but we will not tackle negation directly. Instead, we will first think about *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:

m	$\mathscr{A}$	
n	$\neg\mathscr{A}$	
	$\bot$	$\bot$I m, n

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 sentence first, followed by its negation.

Our elimination rule for '$\bot$' is known as *ex falso quod libet*, or *explosion*. This means 'anything follows from a contradiction', and the idea is precisely that: if we obtained 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.[1] Here is the formal rule:

m	$\bot$	
	$\mathscr{A}$	$\bot$E m

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

A final remark. We have said that '$\bot$' 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 '$\bot$' as a new atomic sentence of TFL, but one which can only ever have the truth value False.
- We might regard '$\bot$' 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 TFL.
- We might regard '$\bot$', not as a symbol of TFL, 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. '$\bot$' 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.

15.10 Negation

There is obviously a tight link between contradiction and negation. Indeed, the $\bot$I rule essentially behaves as a rule for negation elimination: we introduce '$\bot$' when a sentence and its negation both appear in our proof. So there is no need for us to add a further rule for negation elimination.

However, we do need 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:

[1] Thanks to Adam Caulton for this.

m	$\quad \mathscr{A}$	
	$\quad \vdots$	
n	$\quad \bot$	
	$\neg\mathscr{A}$	$\neg$I m–n

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:

1	D	
2	$\quad \neg D$	
3	$\quad \bot$	$\bot$I 1, 2
4	$\neg\neg D$	$\neg$I 2–3

These are all of the basic rules for the proof system for TFL. For ease of reference, they're listed again in appendix C.

Practice exercises

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

1	$\neg L \to (A \wedge L)$	
2	$L \vee \neg L$	LEM
3	$\quad \neg L$	
4	$\quad A$	$\to$E 1, 3
5	$\quad L$	
6	$\quad \bot$	$\bot$I 5, 3
7	$\quad A$	$\bot$E 6
8	A	$\vee$E 2, 3–4, 5–7

1	$A \wedge (B \wedge C)$	
2	$(B \vee C) \to D$	
3	B	$\wedge$E 1
4	$B \vee C$	$\vee$I 3
5	D	$\to$E 4, 2

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.

1	$P \wedge S$
2	$S \to R$
3	P
4	S
5	R
6	$R \vee E$

1	$A \to D$
2	$\quad A \wedge B$
3	$\quad A$
4	$\quad D$
5	$\quad D \vee E$
6	$(A \wedge B) \to (D \vee E)$

1	$\neg L \to (J \vee L)$
2	$\neg L$
3	$J \vee L$
4	$\quad J$
5	$\quad J \wedge J$
6	$\quad J$
7	$\quad L$
8	$\quad \bot$
9	$\quad J$
10	J

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

Additional rules for TFL 16

In §15, we introduced the basic rules of our proof system for TFL. 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*.)

16.1 Disjunctive syllogism

Here is a very natural argument form.

> Elizabeth is 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:

m	$\mathscr{A} \lor \mathscr{B}$	
n	$\neg\mathscr{A}$	
	$\mathscr{B}$	DS m, n

and

m	$\mathscr{A} \lor \mathscr{B}$	
n	$\neg\mathscr{B}$	
	$\mathscr{A}$	DS m, n

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.

16.2 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:

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

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

16.3 Double-negation elimination

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

m	$\neg\neg\mathscr{A}$	
	$\mathscr{A}$	DNE m

We actually gave the argument for this in §**??** 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 TFL forces us to sacrifice certain nuances of English expressions.

16.4 Proof by contradiction

A further rule is very closely related to ¬I. This is *proof by contradiction*, also sometimes called *indirect proof*. It allows us to prove something by assuming its negation and showing that it leads to contradiction. This technique is incredibly common in mathematics, for example it is used to show that $\sqrt{2}$ is irrational.

m	$\quad\mid\ \neg\mathscr{A}$	
n	$\quad\mid\ \bot$	
	$\mathscr{A}$	Contradiction m–n

16.5 De Morgan Rules

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

tables.

The first De Morgan rule is:

m	$\neg(\mathscr{A} \wedge \mathscr{B})$	
	$\neg\mathscr{A} \vee \neg\mathscr{B}$	DeM m

The second De Morgan is the reverse of the first:

m	$\neg\mathscr{A} \vee \neg\mathscr{B}$	
	$\neg(\mathscr{A} \wedge \mathscr{B})$	DeM m

The third De Morgan rule is the *dual* of the first:

m	$\neg(\mathscr{A} \vee \mathscr{B})$	
	$\neg\mathscr{A} \wedge \neg\mathscr{B}$	DeM m

And the fourth is the reverse of the third:

m	$\neg\mathscr{A} \wedge \neg\mathscr{B}$	
	$\neg(\mathscr{A} \vee \mathscr{B})$	DeM m

These are all of the additional rules of our proof system for TFL.

Practice exercises

A. The following proofs are missing their citations (rule and line numbers). Add them wherever they are required:

1	$W \to \neg B$
2	$A \wedge W$
3	$B \vee (J \wedge K)$
4	W
5	$\neg B$
6	$J \wedge K$
7	K

1	$L \leftrightarrow \neg O$
2	$L \vee \neg O$
3	$\quad \neg L$
4	$\quad \neg O$
5	$\quad L$
6	$\quad \bot$
7	$\neg\neg L$
8	L

1	$Z \to (C \land \neg N)$
2	$\neg Z \to (N \land \neg C)$
3	$\quad \neg(N \lor C)$
4	$\quad \neg N \land \neg C$
5	$\quad \neg N$
6	$\quad \neg C$
7	$\quad\quad Z$
8	$\quad\quad C \land \neg N$
9	$\quad\quad C$
10	$\quad\quad \bot$
11	$\quad \neg Z$
12	$\quad N \land \neg C$
13	$\quad N$
14	$\quad \bot$
15	$\neg\neg(N \lor C)$
16	$N \lor C$

B. Give a proof for each of these arguments:

1. $E \lor F$, $F \lor G$, $\neg F \therefore E \land G$
2. $M \lor (N \to M) \therefore \neg M \to \neg N$
3. $(M \lor N) \land (O \lor P)$, $N \to P$, $\neg P \therefore M \land O$
4. $(X \land Y) \lor (X \land Z)$, $\neg(X \land D)$, $D \lor M \therefore M$

Proof-theoretic concepts 17

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 new terminology.

> $\mathscr{A}$ is a THEOREM iff $\vdash \mathscr{A}$

To illustrate this, suppose we want to prove that '$\neg(A \wedge \neg A)$' is a theorem. So we must start our proof without *any* assumptions. However, since we want to prove a sentence whose main logical operator is a negation, we will want to immediately begin a subproof, with the additional assumption '$A \wedge \neg A$', and show that this leads to contradiction. All told, then, the proof looks like this:

1	$\quad\vert\; A \wedge \neg A$	
2	$\quad\vert\; A$	$\wedge$E 1
3	$\quad\vert\; \neg A$	$\wedge$E 1
4	$\quad\vert\; \bot$	$\bot$I 2, 3
5	$\neg(A \wedge \neg A)$	$\neg$I 1–4

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. $O \to O$
2. $N \vee \neg N$
3. $J \leftrightarrow [J \vee (L \wedge \neg L)]$
4. $((A \to B) \to A) \to A$

B. Provide proofs to show each of the following:

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

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.)

Proof strategies 18

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

Work backwards from what you want. The ultimate goal is to obtain the conclusion. Look at the conclusion and ask what the introduction rule is for its main logical operator. This gives you an idea of what should happen *just before* the last line of the proof. Then you can treat this line as if it were your goal. Ask what you could do to get to this new goal.

For example: If your conclusion is a conditional $\mathscr{A} \to \mathscr{B}$, plan to use the $\to$I rule. This requires starting a subproof in which you assume $\mathscr{A}$. The subproof ought to end with $\mathscr{B}$. So, what can you do to get $\mathscr{B}$?

Work forwards from what you have. When you are starting a proof, look at the premises; later, look at the sentences that you have obtained so far. Think about the elimination rules for the main operators of these sentences. These will tell you what your options are.

For a short proof, you might be able to eliminate the premises and introduce the conclusion. A long proof is formally just a number of short proofs linked together, so you can fill the gap by alternately working back from the conclusion and forward from the premises.

Try proceeding indirectly. If you cannot find a way to show $\mathscr{A}$ directly, try starting by assuming $\neg\mathscr{A}$. If a contradiction follows, then you will be able to obtain $\neg\neg\mathscr{A}$ by $\neg$I, and then $\mathscr{A}$ by DNE.

Persist. Try different things. If one approach fails, then try something else.

Derived rules 19

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 §16 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 §16. 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 TFL, but the point is simple: Whatever sentences of TFL 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 other rules of §15. So we did not actually need to introduce it as a *basic* rule in our system; we could have introduced it as a *derived* rule. However, for ease, we have allowed it in as a basic rule.

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 §16, but equally well, you can do this with the *basic* rules of §15:

m	$\mathscr{A} \vee \mathscr{B}$	
n	$\neg\mathscr{A}$	
k	$\quad \mathscr{A}$	
$k+1$	$\quad \bot$	$\bot$I k, n
$k+2$	$\quad \mathscr{B}$	$\bot$E $k+1$
$k+3$	$\quad \mathscr{B}$	
$k+4$	$\quad \mathscr{B}$	R $k+3$
$k+5$	$\mathscr{B}$	$\vee$E m, k–$k+2$, $k+3$–$k+4$

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:

m	$\mathscr{A} \to \mathscr{B}$
n	$\neg\mathscr{B}$

You now want, on line k, to prove $\neg\mathscr{A}$. You can do this with the rule of MT, introduced in §16. Equally well, you can do this with the *basic* rules of §15:

m	$\mathscr{A} \to \mathscr{B}$	
n	$\neg\mathscr{B}$	
k	$\quad \mathscr{A}$	
$k+1$	$\quad \mathscr{B}$	$\to$E m, k
$k+2$	$\quad \bot$	$\bot$I $k+1$, n
$k+3$	$\neg\mathscr{A}$	$\neg$I k–$k+2$

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:

m	$\neg\neg\mathscr{A}$	
j	$\mathscr{A} \vee \neg\mathscr{A}$	LEM
k	$\quad \mathscr{A}$	
$k+1$	$\quad \mathscr{A}$	R k
$k+2$	$\quad \neg\mathscr{A}$	
$k+3$	$\quad \bot$	$\bot$I $k+2$, m
$k+4$	$\quad \mathscr{A}$	$\bot$E $k+3$
$k+5$	$\mathscr{A}$	$\vee$E j, k–$k+1$, $k+2$–$k+4$

Again, we can derive the DNE rule from the *basic* rules of §15.

19.5 Derivation of Proof by Contradiction

Once we have DNE, proof by contradiction is a very quick consequence of ¬I:

m	$\quad \neg\mathscr{A}$	
n	$\quad \bot$	
k	$\neg\neg\mathscr{A}$	$\neg$I m–n
	$\mathscr{A}$	DNE k

19.6 Derivation of De Morgan rules

Here is a demonstration of how we could derive the first De Morgan rule:

m	$\neg(\mathscr{A} \wedge \mathscr{B})$	
j	$\mathscr{A} \vee \neg\mathscr{A}$	LEM
k	$\quad \mathscr{A}$	
$k+1$	$\quad\quad \mathscr{B}$	
$k+2$	$\quad\quad \mathscr{A} \wedge \mathscr{B}$	$\wedge$I k, $k+1$
$k+3$	$\quad\quad \bot$	$\bot$I $k+2$, m
$k+4$	$\quad \neg\mathscr{B}$	$\neg$I $k+1$–$k+3$
$k+5$	$\quad \neg\mathscr{A} \vee \neg\mathscr{B}$	$\vee$I $k+4$
$k+6$	$\quad \neg\mathscr{A}$	
$k+7$	$\quad \neg\mathscr{A} \vee \neg\mathscr{B}$	$\vee$I $k+6$
$k+8$	$\neg\mathscr{A} \vee \neg\mathscr{B}$	$\vee$E j, k–$k+5$, $k+6$–$k+7$

Here is a demonstration of how we could derive the second De Morgan rule:

m	$\neg\mathscr{A} \vee \neg\mathscr{B}$	
k	$\quad \mathscr{A} \wedge \mathscr{B}$	
$k+1$	$\quad \mathscr{A}$	$\wedge$E k
$k+2$	$\quad \mathscr{B}$	$\wedge$E k
$k+3$	$\quad\quad \neg\mathscr{A}$	
$k+4$	$\quad\quad \bot$	$\bot$I $k+1$, $k+3$
$k+5$	$\quad\quad \neg\mathscr{B}$	
$k+6$	$\quad\quad \bot$	$\bot$I $k+2$, $k+5$
$k+7$	$\quad \bot$	$\vee$E m, $k+3$–$k+4$, $k+5$–$k+6$
$k+8$	$\neg(\mathscr{A} \wedge \mathscr{B})$	$\neg$I k–$k+7$

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 §§16–17 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.)

Soundness and completeness 20

In §17, 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, $\text{valid}_{\vDash}$ and $\text{valid}_{\vdash}$ 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. 89). Since the negation of a contradiction is a tautology, we can define a SYNTACTIC CONTRADICTION IN TFL 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 TFL if it is not a theorem or a contradiction.

A collection of sentences are PROVABLY INCONSISTENT IN TFL 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 TFL if and only if they are not provably inconsistent.

Finally, an argument is PROVABLY VALID IN TFL 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 $\text{valid}_{\vdash}$ implies $\text{valid}_{\vDash}$. Afterwards, we will need to show things in the other directions, $\text{valid}_{\vDash}$ implies $\text{valid}_{\vdash}$

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 TFL using a recursive definition (see p. 30). We could have also used recursive definitions to define a proper proof in TFL 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 TFL. 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 TFL 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 connectives (see p. 30). 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 TFL, we can sketch how a recursive proof of the soundness of TFL 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 $\mathscr{A}, \mathscr{B} \vdash \mathscr{A} \wedge \mathscr{B}$; $\mathscr{A} \wedge \mathscr{B} \vdash \mathscr{A}$; $\mathscr{A} \vee \mathscr{B}, \neg\mathscr{A} \vdash \mathscr{B}$ ... etc. Since some rules have a couple different forms, we would have to have add some members to this base class, for instance $\mathscr{A} \wedge \mathscr{B} \vdash \mathscr{B}$ Notice that these are all statements in the metalanguage. The proof that TFL is sound is not a part of TFL, because TFL 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 $\text{valid}_{\vDash}$. For instance the proof $\mathscr{A}, \mathscr{B} \vdash \mathscr{A} \wedge \mathscr{B}$ corresponds to a truth table that shows $\mathscr{A}, \mathscr{B} \vDash \mathscr{A} \wedge \mathscr{B}$ This establishes the first part of our recursive proof.

The next step is to show that adding lines to any proof will never change

Concept	**Truth table (semantic) definition**	**Proof-theoretic (syntactic) definition**
Tautology	A sentence whose truth table only has Ts under the main connective	A sentence that can be derived without any premises.
Contradiction	A sentence whose truth table only has Fs under the main connective	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 connective	A sentence that is not a theorem or contradiction
Equivalent sentences	The columns under the main connectives are identical.	The sentences can be derived from each other
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
Consistent sentences	Sentences which have at least one line in their truth table where they are all true.	Sentences which are not inconsistent
Valid argument	An argument whose truth table has no lines where there are all Ts under main connectives for the premises and an F under the main connective for the conclusion.	An argument where one can derive the conclusion from the premises

Table 20.1: Two ways to define logical concepts.

a valid$_{\vDash}$ 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 TFL is both sound and complete. This is not the case for all proof systems or all formal languages. Because it is true of TFL, 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 tautology	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 $A \to [((B \wedge C) \vee D) \to A]$ is a tautology.
2. Show that $A \to (A \to B)$ is not a tautology
3. Show that the sentence $A \to \neg A$ is not a contradiction.
4. Show that the sentence $A \leftrightarrow \neg A$ is a contradiction.
5. Show that the sentence $\neg(W \to (J \vee J))$ 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 $B \to \neg S$ is equivalent to the sentence $\neg\neg B \to \neg S$
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, $C \to A$ are jointly inconsistent.
10. Show that the sentences $\neg(A \vee B)$, $\neg B$, $B \to A$ 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 $A \to (B \to A)$ is a tautology
2. Show that $\neg(((N \leftrightarrow Q) \vee Q) \vee N)$ is not a tautology
3. Show that $Z \vee (\neg Z \leftrightarrow Z)$ is contingent
4. show that $(L \leftrightarrow ((N \to N) \to L)) \vee H$ 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 $\neg(W \to W)$, $(W \leftrightarrow W) \wedge W$, $E \vee (W \to \neg(E \wedge W))$ are inconsistent.
10. Show that the sentences $\neg R \vee C$, $(C \wedge R) \to \neg R$, $(\neg(R \vee R) \to R)$ are consistent.
11. Show that $\neg\neg(C \leftrightarrow \neg C), ((G \vee C) \vee G) \therefore ((G \to C) \wedge G)$ is valid.
12. Show that $\neg\neg L, (C \to \neg L) \to C) \therefore \neg C$ is invalid.

Chapter 5

First-order logic

Building blocks of FOL 21

21.1 The need to decompose sentences

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

1. Willard is a logician.
2. All logicians wear funny hats.

$\therefore$ Willard wears a funny hat.

To symbolize it in TFL, 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 TFL, 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 TFL*. The problem lies with TFL 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 TFL are atomic sentences, and TFL 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 *first-order logic*, or *FOL*.

The details of FOL will be explained throughout this chapter, but here is the basic idea for splitting the atom.

First, we have *names*. In FOL, 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 FOL, we indicate predicates with uppercase italic letters. For instance, we might let the FOL predicate 'D' symbolize the English predicate '_____ is a dog'. Then the expression 'Db' will be a sentence in FOL, which symbolizes the English sentence 'Bertie is a dog'. Equally, we might let the FOL 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 FOL 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 FOL is significantly more subtle than TFL, 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 FOL, 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 FOL:

$$a, b, c, \ldots, r, a_1, f_{32}, j_{390}, m_{12}$$

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 FOL, 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 TFL, 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 FOL, PREDICATES are capital letters A through Z, with or without subscripts. We might write a symbolization key for predicates thus:

Ax: $______x$ is angry
Hx: $______x$ is happy

(Why the subscripts on the gaps? We will return to this in the next section.)

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 TFL, and symbolize the entire sentence by '$Ag \wedge Am$'. This illustrates an important point: FOL has all of the truth-functional connectives of TFL.

Sentence 3: this is a conditional, whose antecedent is sentence 1 and whose consequent is sentence 2, so we can symbolize this with '$Ae \rightarrow (Ag \wedge Am)$'.

Sometimes the truth functional nature of a sentence isn't explicit. For example suppose we wish to symbolise:

4. Betty is a red car.

This sentence can be paraphrased as 'Betty is red and Betty is a car'. And we can symbolise this as $Rb \wedge Cb$ using the symbolisation key

b: Betty
Rx: $______x$ is red
Cx: $______x$ is a car

21.4 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:

5. Karl loves Imre.
6. Imre loves himself.
7. Karl loves Imre, but not vice versa.
8. Karl is loved by Imre.

We will start with the following representation key:

domain: people
i: Imre
k: Karl
Lxy: _____$_x$ loves _____$_y$

Sentence 5 will now be symbolized by 'Lki'.

Sentence 6 can be paraphrased as 'Imre loves Imre'. It can now be symbolized by 'Lii'.

Sentence 7 is a conjunction. We might paraphrase it as 'Karl loves Imre, and Imre does not love Karl'. It can now be symbolized by '$Lki \wedge \neg Lik$'.

Sentence 8 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 FOL.

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 (just 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.

21.5 Quantifiers

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

9. Everyone is happy.
10. Someone is angry.

It might be tempting to symbolize sentence 9 as '$He \wedge Hg \wedge Hm$'. 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 FOL, variables are italic lowercase letters 's' through 'z', with or without subscripts. So we might symbolize sentence 9 as '$\forall xHx$'. 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 '$\forall xHx$', '$\forall yHy$', '$\forall zHz$', and '$\forall x_5Hx_5$' use different variables, but they will all be logically equivalent.

To symbolize sentence 10, we introduce another new symbol: the EXISTENTIAL QUANTIFIER, '$\exists$'. Like the universal quantifier, the existential quantifier requires a variable. Sentence 10 can be symbolized by '$\exists xAx$'. Whereas '$\forall xAx$' is read naturally as 'for all x, x is angry', '$\exists xAx$' 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 10 with '$\exists z Az$', '$\exists w_{256} Aw_{256}$', or whatever.

Some more examples will help. Consider these further sentences:

11. No one is angry.
12. There is someone who is not happy.
13. Not everyone is happy.

Sentence 11 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 11 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 12 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'. That too would be a perfectly adequate symbolization of sentence 13.

21.6 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 FOL, the domain must always include at least one thing. Moreover, in English we can infer 'something is angry' from 'Gregor is angry'. In FOL, 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 FOL, 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:

14. Chimera is angry.
15. 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 14 as Ac and sentence 15 as $\neg Ac$.

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

One option is to say that sentence 14 is not true, because there is no chimera. If sentence 14 is false because it talks about a non-existent thing, then sentence 15 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 14 is *meaningless* because it talks about a non-existent thing. So Ac would be a meaningful expression in FOL 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. 173.) The important point for now is that each name of FOL *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.

Sentences with one quantifier 22

We now have all of the pieces of FOL. 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: $______x$ is in my pocket
Tx: $______x$ is on the table
Qx: $______x$ is a quarter
Dx: $______x$ 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 TFL, $\mathscr{A} \rightarrow \mathscr{B}$ is logically equivalent (in TFL) to $\neg\mathscr{A} \vee \mathscr{B}$. This equivalence will also hold in FOL. 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 FOL 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 FOL. 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 '$\forall x(Rx \to Tx)$', since '$\forall x Tx$' would symbolize the sentence 'every person or plant has a thorn'. Similarly, we will have to symbolize sentence 8 with '$\forall x(Cx \to Sx)$'.

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 FOL, it is important to understand the structure of the sentences you want to symbolize. What matters is the final symbolization in FOL, and sometimes you will be able to move from an English language sentence directly to a sentence of FOL. 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 FOL.

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 '$\forall xBx \rightarrow Bl$'. 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 '$\forall x(Bx \rightarrow Bl)$'. 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 TFL, 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.

22.6 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:

$$\begin{array}{ll} & \forall x\big[\neg(Rx \land Kx) \to \neg Hx\big] \\ & \forall x(Rx \to Gx) \\ & Rb \land \neg Kb \\ \therefore & Gb \land \neg Hb \end{array}$$

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:

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

This symbolization is a disaster! It takes what in English is a terrible argument and symbolizes it as a valid argument in FOL. 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:

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

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 FOL.

B. Using the following symbolization key:

domain: people
Kx: $______x$ knows the combination to the safe
Sx: $______x$ is a spy
Vx: $______x$ is a vegetarian
h: Hofthor
i: Ingmar

symbolize the following sentences in FOL:

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: $______x$ is an alligator.
Mx: $______x$ is a monkey.
Rx: $______x$ is a reptile.
Zx: $______x$ lives at the zoo.
a: Amos
b: Bouncer
c: Cleo

symbolize each of the following sentences in FOL:

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 FOL.

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.

Multiple quantifiers 23

We see more power of FOL when quantifiers start stacking on top of one another.

Consider

1. Someone loves everyone.

Before considering how to symbolise that, we start of by symbolising the related sentence:

2. John loves everyone.

This can be symbolised as $\forall x Ljx$, using the symbolisation key:

domain: all people
j: John
Lxy: $______x$ loves $______y$

This gives us an insight into how to symbolise 1: it's like 2 except it might not be John who loves everyone, 1 just says that there is *someone*, y, such that y loves everyone. We will thus symbolise it $\exists y \forall x Lyx$.

In the earlier examples we always used x as our variables; but we here had to use y because x is already taken. We don't want to say $\exists x \forall x Lxx$ as it's not clear how one should read this sentence as we need to identify which variables come with which quantifiers.

23.1 The order of quantifiers

Consider the sentence 'everyone loves someone'. This is potentially ambiguous. It might mean either of the following:

3. For every person x, there is some person that x loves
4. There is some particular person whom every person loves

Sentence 3 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 3 is true.

Sentence 4 is symbolized by '$\exists y \forall x Lxy$'. Sentence 4 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. ($\forall\exists$)
$\therefore$ 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. ($\forall\exists$)
$\therefore$ There is a day for all the dogs. ($\exists\forall$)

The moral is: take great care with the order of quantification.

23.2 Stepping-stones to symbolization

Once we have the possibility of multiple quantifiers, representation in FOL 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:

5. Geraldo is a dog owner.
6. Someone is a dog owner.
7. All of Geraldo's friends are dog owners.
8. Every dog owner is a friend of a dog owner.
9. Every dog owner's friend owns a dog of a friend.

Sentence 5 can be paraphrased as, 'There is a dog that Geraldo owns'. This can be symbolized by '$\exists x(Dx \wedge Ogx)$'.

Sentence 6 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 5, except that it is not specifically about Geraldo. So we can symbolize sentence 6 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 FOL sentence *nor* an English sentence: it uses bits of FOL ('$\exists$', '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 FOL sentence. You

[1] Thanks to Rob Trueman for the example.

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 7 can be paraphrased as, 'Everyone who is a friend of Geraldo is a dog owner'. Using our stepping-stone tactic, we might write

$$\forall x\big[Fxg \to x \text{ is a dog owner}\big]$$

Now the fragment that we have left to deal with, 'x is a dog owner', is structurally just like sentence 5. However, it would be a mistake for us simply to write

$$\forall x\big[Fxg \to \exists x(Dx \land Oxx)\big]$$

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:

$$\forall x\big[Fxg \to \exists z(Dz \land Oxz)\big]$$

This adequately symbolizes sentence 7.

Sentence 8 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

$$\forall x\big[x \text{ is a dog owner} \to \exists y(y \text{ is a dog owner} \land Fxy)\big]$$

Completing the symbolization, we end up with

$$\forall x\big[\exists z(Dz \land Oxz) \to \exists y\big(\exists z(Dz \land Oyz) \land Fxy\big)\big]$$

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:

$$\overbrace{\forall x\big[\overbrace{\exists z(Dz \land Oxz)}^{\text{scope of 1st '}\exists z\text{'}} \to \overbrace{\exists y\big(\overbrace{\exists z(Dz \land Oyz)}^{\text{scope of 2nd '}\exists z\text{'}} \land Fxy\big)}^{\text{scope of '}\exists y\text{'}}\big]}^{\text{scope of '}\forall x\text{'}}$$

This shows that no variable is being forced to serve two masters simultaneously.

Sentence 9 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:

$$\forall x\big[x \text{ is a friend of a dog owner} \to x \text{ owns a dog which is owned by a friend of } x\big]$$

Breaking this down a bit more:

$$\forall x\big[\exists y(Fxy \wedge y \text{ is a dog owner}) \rightarrow \\ \exists y(Dy \wedge Oxy \wedge y \text{ is owned by a friend of } x)\big]$$

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!

23.3 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 FOL 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,

[2]The Salique law was the common law of France which prohibited the crown be passed on to female heirs.

> they might have known that *One* Woman is superior to *All* the Men in these Nations, or 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.,

$$\forall x(Mx \to \forall y(Wy \to Sxy))$$

or that some men are superior to some women,

$$\exists x(Mx \land \exists y(Wy \land Sxy)).$$

The latter is true, but so is

$$\exists y(Wy \land \exists x(Mx \land Syx)).$$

(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.,

$$\exists y(Wy \land \forall x(Mx \to Syx)).$$

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 FOL:

1. If Cleo loves Bouncer, then Bouncer is a monkey.
2. If both Bouncer and Cleo are alligators, then Amos loves them both.
3. Cleo loves a reptile.
4. Bouncer loves all the monkeys that live at the zoo.

[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.

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 FOL:

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 FOL.

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 FOL:

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 FOL:

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.

Sentences of FOL 24

We know how to represent English sentences in FOL. The time has finally come to define the notion of a *sentence* of FOL.

24.1 Expressions

There are six kinds of symbols in FOL:

Predicates $A, B, C, \ldots, Z$, with subscripts, as needed: $A_1, Z_2, A_{25}, J_{375}, \ldots$.[1]

Names $a, b, c, \ldots, r$, or with subscripts, as needed $a_1, b_{224}, h_7, m_{32}, \ldots$

Variables s, t, u, v, w, x, y, z, or with subscripts, as needed $x_1, y_1, z_1, x_2, \ldots$

Connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Brackets (,)

Quantifiers $\forall, \exists$

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

But which expressions are sentences?

As we did for TFL, we will present a recursive definition. The starting point of this is *atomic sentences*. In TFL, these were given to us. In FOL, we form atomic sentences using predicates and names. For example Aa will be an atomic sentence, so will Rab. We will express this in general using script fonts for metavariables, as introduced in §7:

> If $\mathscr{R}$ is an n-place predicate and $\mathscr{a}_1, \mathscr{a}_2, \ldots, \mathscr{a}_n$ are names, then $\mathscr{R}\mathscr{a}_1\mathscr{a}_2 \ldots \mathscr{a}_n$ is an atomic sentence.

[1] Each predicate will have a number of places associated with it. We should thus really introduce:

One-Place Predicates $A^1, B^1, \ldots, Z^1$, with subscripts, as needed: $A^1_1, Z^1_2, A^1_{25}, J^1_{375}, \ldots$

Two-Place Predicates $A^2, B^2, \ldots, Z^2$, with subscripts, as needed: $A^2_1, Z^2_2, A^2_{25}, J^2_{375}, \ldots$

Three-Place Predicates $A^3, B^3, \ldots, Z^3$, with subscripts, as needed: $A^3_1, Z^3_2, A^3_{25}, J^3_{375}, \ldots$

etc. We drop the superscripts for ease.

We can now give the recursive definition of what it is to be a sentence of FOL:

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,
   - ▷ x is a variable that does not appear in $\mathscr{A}$,
   - ▷ $\mathscr{A}'$ is the expression that is the result of replacing any occurrences of a name, a, by x.

   then $\exists x\mathscr{A}'$ and $\forall x\mathscr{A}'$
8. Nothing else is a sentence.

The interesting clause here is 7. Let's see how it is used in action. Using the first clause we can see that Aa is a sentence. 7 then lets us replace a by x, and *bind* this with a quantifier, for example $\forall xAx$.

We can then proceed to see that $(\forall xAx \wedge Ab)$ is a sentence using the atomic clause and the clause for $\wedge$. We can then replace the name b with a variable y and bind it with an existential quantifier, finding the sentence $\exists y(\forall xAx \wedge Ay)$

For more complicated sentences, we can display the structure and construction of the sentence perspicuously by presenting a formation tree, similarly to as we did in 6.2.

$$\begin{array}{c}
\exists z\forall y(Ryz \wedge \exists xQx) \\
| \\
\forall y(Ryc \wedge \exists xQx) \\
| \\
(Rbc \wedge \exists xQx) \\
/ \quad \backslash \\
Rbc \quad \exists xQx \\
| \\
Qa
\end{array}$$

Moving up the formation tree is following one of the rules of the recursive specification of what it is to be a sentence.

Why in 7 did we have the restriction that x is not already in $\mathscr{A}$? This is to ensure that variables only serve one master at any one time (see §23). Otherwise

we could see that $\forall x Rax$ is a sentence and then conclude that $\exists x \forall x Rxx$, which is not a good sentence.

In fact, 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 TFL, though we note that a logical operator can be either a connective or a quantifier:

> The MAIN LOGICAL OPERATOR in a formula is the operator that was introduced last, when that formula was constructed using the recursion rules.
>
> The SCOPE of a logical operator in a formula is the subformula for which that operator is the main logical operator.

So we can graphically illustrate the scope of the quantifiers in the preceding example thus:

$$\overbrace{\forall x\, \overbrace{\exists y(Fx \leftrightarrow \overbrace{\forall z(Gayz \rightarrow Syzyayx)}^{\text{scope of '}\forall z\text{'}})}^{\text{scope of '}\exists y\text{'}}}^{\text{scope of '}\forall x\text{'}}$$

24.2 Bracketing conventions

We will adopt the same notational conventions governing brackets that we did for TFL (see §6 and §10.3.)

First, we may omit the outermost brackets of a formula.

Second, we may use square brackets, '[' and ']', in place of brackets to increase the readability of formulas.

Practice exercises

A. Identify which variables are bound and which are free.

1. $\exists x Lxy \wedge \forall y Lyx$
2. $\forall x Ax \wedge Bx$
3. $\forall x(Ax \wedge Bx) \wedge \forall y(Cx \wedge Dy)$
4. $\forall x \exists y[Rxy \rightarrow (Jz \wedge Kx)] \vee Ryx$
5. $\forall x_1(Mx_2 \leftrightarrow Lx_2x_1) \wedge \exists x_2 Lx_3x_2$

Chapter 6

Interpretations

Extensionality 25

Recall that TFL is a truth-functional language. Its connectives are all truth-functional, and *all* that we can do with TFL is key sentences to particular truth values. We can do this *directly*. For example, we might stipulate that the TFL 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 TFL 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 TFL cannot handle differences in meaning that go beyond mere differences in truth value.

25.1 Symbolizing versus translating

FOL 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 FOL predicates, such as:

Cx: $______x$ teaches Logic III in Calgary

we do not carry the *meaning* of the English predicate across into our FOL 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 FOL does not give us any resources for dealing with nuances of meaning. When we interpret FOL, 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 FOL is an EXTENSIONAL LANGUAGE because FOL does not represent differences of meaning between predicates that have the same extension.

For this reason, we say only that FOL sentences *symbolize* English sentences. It is doubtful that we are *translating* English into FOL, 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' here are symbols of augmented English, and that they are being *used*. By contrast, 'x' and 'y' are symbols of FOL, 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 TFL as any assignment of truth and falsity to atomic sentences. In FOL, we are going to define an INTERPRETATION as consisting of three things:

- the specification of a domain
- for each name that we care to consider, an assignment of exactly one object within the domain
- for each predicate that we care to consider—other than '='—a specification of what things (in what order) the predicate is to be true of

The symbolization keys that we considered in Part 5 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:

$$\begin{tikzcd}
1 \arrow[loop left] & 2 \\
4 & 3 \arrow[l] \arrow[ul, leftrightarrow] \arrow[loop right]
\end{tikzcd}$$

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.

Truth in FOL 26

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 FOL sentence to be true or false in an interpretation.

We know from §24 that there are three kinds of sentence in FOL:

- atomic sentences
- sentences whose main logical operator is a sentential connective
- sentences whose main logical 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. 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 $\mathscr{a}_1, \mathscr{a}_2, \ldots, \mathscr{a}_n$ are names,

> $\mathscr{R}\mathscr{a}_1\mathscr{a}_2\ldots\mathscr{a}_n$ is true in an interpretation **iff**
> $\mathscr{R}$ is true of the objects named by $\mathscr{a}_1, \mathscr{a}_2, \ldots, \mathscr{a}_n$ in that interpretation (considered in that order)

26.2 Sentential connectives

We saw in §24 that FOL sentences can be built up from simpler ones using the truth-functional connectives that were familiar from TFL. The rules governing these truth-functional connectives are *exactly* the same as they were when we considered TFL. 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 connectives; 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
- '$\neg Pa$' is false
- '$Pa \wedge \neg(Pb \wedge Rab)$' is true, because '$Pa$' is true and Pb is false, so $Pb \wedge Rab$ is false, thus $\neg(Pb \wedge Rab)$ is also true.

Make sure you understand these examples.

26.3 When the main logical operator is a quantifier

The exciting innovation in FOL, though, is the use of *quantifiers*. But expressing the truth conditions for quantified sentences is a bit more fiddly than one might first expect.

Here is a naïve first thought. We want to say that '$\forall x Fx$' is true iff 'Fx' is true of everything in the domain. This should not be too problematic: our interpretation will specify directly what 'Fx' is true of.

Unfortunately, this naïve thought is not general enough. For example, 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.

So here is a second 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 either. 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 a third 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. So we should focus on this possibility of extending an interpretation, by adding a new name. We will offer a few examples of how this might work, centring 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 extend our go-to interpretation—temporarily, mind—by adding the name 'c' to refer to Lady Gaga, then 'Rbc' would be true on this extended interpretation. This, surely, should 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 extend our go-to interpretation by letting a new name, 'c', denote Socrates, then '$Wc \wedge Rca$' would be true on this extended interpretation. Again, this should surely suffice to make '$\exists x(Px \wedge Rxa)$' true on the original go-to 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 extend our go-to interpretation by letting a new name, 'd', denote that person, then we would not be able to find anyone else in the domain to denote with some further new name, perhaps 'e', in such a way that 'Rde' would be true. Indeed, no matter *whom* we named with 'e', 'Rde' would be false. This observation is surely sufficient to make '$\exists y Rdy$' *false* in our extended interpretation, which in turn is surely sufficient to make '$\forall x \exists y Rxy$' false on the original go-to 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!

Suppose that $\mathscr{A}$ is an expression containing at least one instance of the variable $\mathscr{x}$, and that $\mathscr{x}$ is free in $\mathscr{A}$. We will write this thus:

$$\mathscr{A}(\ldots \mathscr{x} \ldots \mathscr{x} \ldots)$$

Suppose also that $\mathscr{c}$ is a name. Then we will write:

$$\mathscr{A}(\ldots \mathscr{c} \ldots \mathscr{c} \ldots)$$

for the formula obtained by replacing every 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 $\exists \mathscr{x}\mathscr{A}$. $\mathscr{c}$ is called the INSTANTIATING NAME. So:

$$\exists x(Rex \leftrightarrow Fx)$$

is a substitution instance of

$$\forall y \exists x (Ryx \leftrightarrow Fx)$$

with the instantiating name 'e'.

Armed with this notation, the rough idea is as follows. The sentence $\forall \boldsymbol{x} \mathscr{A}(\ldots \boldsymbol{x} \ldots \boldsymbol{x} \ldots)$ will be true iff $\mathscr{A}(\ldots \boldsymbol{c} \ldots \boldsymbol{c} \ldots)$ is true no matter what object (in the domain) we name with $\boldsymbol{c}$. Similarly, the sentence $\exists \boldsymbol{x} \mathscr{A}$ will be true iff there is *some* way to assign the name $\boldsymbol{c}$ to an object that makes $\mathscr{A}(\ldots \boldsymbol{c} \ldots \boldsymbol{c} \ldots)$ true. More precisely, we stipulate:

> $\forall \boldsymbol{x} \mathscr{A}(\ldots \boldsymbol{x} \ldots \boldsymbol{x} \ldots)$ is true in an interpretation **iff**
> $\mathscr{A}(\ldots \boldsymbol{c} \ldots \boldsymbol{c} \ldots)$ is true in *every* interpretation that extends the original interpretation by assigning an object to any name $\boldsymbol{c}$ (without changing the interpretation in any other way).
>
> $\exists \boldsymbol{x} \mathscr{A}(\ldots \boldsymbol{x} \ldots \boldsymbol{x} \ldots)$ is true in an interpretation **iff**
> $\mathscr{A}(\ldots \boldsymbol{c} \ldots \boldsymbol{c} \ldots)$ is true in *some* interpretation that extends the original interpretation by assigning an object to some name $\boldsymbol{c}$ (without changing the interpretation in any other way).

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.

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. $Nc \rightarrow (Ac \vee Bc)$
4. $\forall x Ax$
5. $\forall x \neg Bx$
6. $\exists x (Ax \wedge Bx)$
7. $\exists x (Ax \rightarrow Nx)$
8. $\forall x (Nx \vee \neg Nx)$
9. $\exists x Bx \rightarrow \forall x Ax$

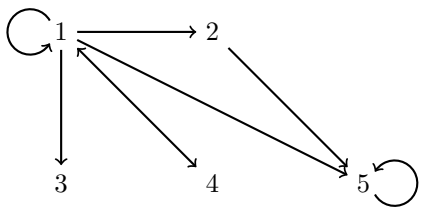
B. Consider the following interpretation:

- 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. $Mc \vee Me$
4. $Gc \vee \neg Gc$
5. $Mc \rightarrow Gc$
6. $\exists x Hx$
7. $\forall x Hx$
8. $\exists x \neg Mx$
9. $\exists x(Hx \wedge Gx)$
10. $\exists x(Mx \wedge Gx)$
11. $\forall x(Hx \vee Mx)$
12. $\exists x Hx \wedge \exists x Mx$
13. $\forall x(Hx \leftrightarrow \neg Mx)$
14. $\exists x Gx \wedge \exists x \neg Gx$
15. $\forall x \exists y(Gx \wedge Hy)$

C. Following the diagram conventions introduced at the end of §25, consider the following interpretation:



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. $\forall x(\exists y Rxy \rightarrow \exists y Ryx)$
9. $\exists x \exists y(\neg x = y \wedge Rxy \wedge Ryx)$
10. $\exists x \forall y(Rxy \leftrightarrow x = y)$
11. $\exists x \forall y(Ryx \leftrightarrow x = y)$
12. $\exists x \exists y(\neg x = y \wedge Rxy \wedge \forall z(Rzx \leftrightarrow y = z))$

Semantic concepts 27

Offering a precise definition of truth in FOL 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 TFL. However, remember that they concern *interpretations*, rather than valuations.

We will use the symbol '$\vDash$' for FOL much as we did for TFL. So:

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

means that there is no interpretation in which all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are true and in which $\mathscr{C}$ is false. Derivatively,

$$\vDash \mathscr{A}$$

means that $\mathscr{A}$ is true in every interpretation.

The other logical notions also have corresponding definitions in FOL:

- ▷ An FOL sentence $\mathscr{A}$ is a LOGICAL TRUTH iff $\mathscr{A}$ is true in every interpretation; i.e., $\vDash \mathscr{A}$.
- ▷ $\mathscr{A}$ is a CONTRADICTION 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 FOL 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 FOL otherwise.
- ▷ Two FOL sentences $\mathscr{A}$ and $\mathscr{B}$ are LOGICALLY EQUIVALENT 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 FOL sentences $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are JOINTLY CONSISTENT iff there is some interpretation in which all of the sentences are true. They are JOINTLY INCONSISTENT iff there is no such interpretation.

Using interpretations 28

28.1 Logical truths and contradictions

Suppose we want to show that '$\exists xAxx \rightarrow Bd$' is *not* a logical truth. 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 logical truth.

In order for '$\exists xAxx \rightarrow Bd$' to be false, the antecedent ('$\exists xAxx$') 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 '$\exists xAxx$' 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 '$\exists xAxx$'. 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 '$\exists xAxx$' is true, but where 'Bd' is false. So there is an interpretation where '$\exists xAxx \rightarrow Bd$' is false. So '$\exists xAxx \rightarrow Bd$' is not a logical truth.

We can just as easily show that '$\exists xAxx \rightarrow Bd$' is not a contradiction. We need only specify an interpretation in which '$\exists xAxx \rightarrow Bd$' is true; i.e., an interpretation in which either '$\exists xAxx$' 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 '$\exists xAxx \to Bd$' is true. So '$\exists xAxx \to Bd$' is not a contradiction.

28.2 Logical equivalence

Suppose we want to show that '$\forall xSx$' and '$\exists xSx$' are not logically 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 '$\exists xSx$' true by including something in the extension of 'S', and we can make '$\forall xSx$' false by leaving something out of the extension of 'S'. For concreteness we will offer:

Sx: $______x$ plays saxophone

Now '$\exists xSx$' is true, because 'Sx' is true of Ornette Coleman. Slightly more precisely, extend our interpretation by allowing 'c' to name Ornette Coleman. 'Sc' is true in this extended interpretation, so '$\exists xSx$' was true in the original interpretation. Similarly, '$\forall xSx$' is false, because 'Sx' is false of Miles Davis. Slightly more precisely, extend our interpretation by allowing 'd' to name Miles Davis, and 'Sd' is false in this extended interpretation, so '$\forall xSx$' was false in the original interpretation. We have provided a counter-interpretation to the claim that '$\forall xSx$' and '$\exists xSx$' are logically equivalent.

> To show that $\mathscr{A}$ is not a logical truth, 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 logically equivalent, it suffices to find an interpretation where one is true and the other is false.

28.3 Validity, entailment and consistency

To test for validity, entailment, or consistency, we typically need to produce interpretations that determine the truth value of several sentences simultaneously.

Consider the following argument in FOL:

$$\exists x(Gx \to Ga) \therefore \exists xGx \to Ga$$

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: $_____{}_x$ 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 logical truth.) 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 consistent.

Let's consider a second example. Consider:

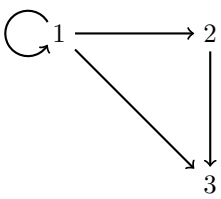
$$\forall x \exists y Lxy \therefore \exists y \forall x Lxy$$

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: $_____{}_x$ is in a civil partnership with $_____{}_y$

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 consistent 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:



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
- '$\exists x \forall y(Ryx \leftrightarrow x = y)$' witness 1
- '$\exists x \exists y \exists z((\neg y = z \wedge Rxy) \wedge Rzx)$' witness 2

- '$\exists x \forall y \neg Rxy$' witness 3
- '$\exists x(\exists y Ryx \wedge \neg \exists y Rxy)$' witness 3

This immediately shows that all of the preceding six sentences are jointly consistent. We can use this observation to generate *invalid* arguments, e.g.:

$$\forall x \exists y Ryx, \exists x \forall y Rxy \therefore \forall x \exists y Rxy$$
$$\exists x \forall y Rxy, \exists x \forall y \neg Rxy \therefore \neg \exists x \exists y \exists z(\neg y = z \wedge (Rxy \wedge Rzx))$$

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 consistent.

When you provide an interpretation to refute a claim—to logical truth, say, or to entailment—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 logical truth 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. $\exists x(Gx \rightarrow \forall y My)$
7. $\exists x(x = h \wedge x = i)$

B. Show that the following pairs of sentences are not logically 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 consistent:

1. $Ma, \neg Na, Pa, \neg Qa$
2. $Lee, Leg, \neg Lge, \neg Lgg$
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. $\forall y Gy, \forall x(Gx \to Hx), \exists y \neg Iy$
6. $\exists x(Bx \vee Ax), \forall x \neg Cx, \forall x\big[(Ax \wedge Bx) \to Cx\big]$
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. $\exists z(Nz \wedge Ozz), \forall x \forall y(Oxy \to Oyx)$
10. $\neg \exists x \forall y Rxy, \forall x \exists y Rxy$
11. $\neg Raa$, $\forall x(x = a \vee Rxa)$
12. $\forall x \forall y \forall z[(x = y \vee y = z) \vee x = z]$, $\exists x \exists y\ \neg x = y$
13. $\exists x \exists y((Zx \wedge Zy) \wedge x = y)$, $\neg Zd$, $d = e$

D. Show that the following arguments are invalid:

1. $\forall x(Ax \to Bx) \therefore \exists x Bx$
2. $\forall x(Rx \to Dx), \forall x(Rx \to Fx) \therefore \exists x(Dx \wedge Fx)$
3. $\exists x(Px \to Qx) \therefore \exists x Px$
4. $Na \wedge Nb \wedge Nc \therefore \forall x Nx$
5. $Rde, \exists x Rxd \therefore Red$
6. $\exists x(Ex \wedge Fx), \exists x Fx \to \exists x Gx \therefore \exists x(Ex \wedge Gx)$
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. $Lab \to \forall x Lxb, \exists x Lxb \therefore Lbb$
10. $\forall x(Dx \to \exists y Tyx) \therefore \exists y \exists z\ \neg y = z$

Reasoning about all interpretations 29

29.1 Logical truths and contradictions

We can show that a sentence is *not* a logical truth just by providing one carefully specified interpretation: an interpretation in which the sentence is false. To show that something is a logical truth, 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 logical truth 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 '$Raa \leftrightarrow Raa$' is a logical truth:

> Any relevant interpretation will give 'Raa' a truth value. If 'Raa' is true in an interpretation, then '$Raa \leftrightarrow Raa$' is true in that interpretation. If 'Raa' is false in an interpretation, then '$Raa \leftrightarrow Raa$' is true in that interpretation. These are the only alternatives. So '$Raa \leftrightarrow Raa$' is true in every interpretation. Therefore, it is a logical truth.

This argument is valid, of course, and its conclusion is true. However, it is not an argument in FOL. Rather, it is an argument in English *about* FOL: 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 TFL sentences.)

Here is another bit of reasoning. Consider the sentence '$\forall x(Rxx \leftrightarrow Rxx)$'. Again, it should obviously be a logical truth, but to say precisely why is quite a challenge. We cannot say that '$Rxx \leftrightarrow Rxx$' is true in every interpretation, since '$Rxx \leftrightarrow Rxx$' is not even a *sentence* of FOL (remember that 'x' is a variable, not a name). So we have to be a bit cleverer.

> Consider some arbitrary interpretation. Consider some arbitrary member of the domain, which, for convenience, we will call *obbie*, and suppose we extend our original interpretation by adding a new

> name, 'c', to name *obbie*. Then either 'Rcc' will be true or it will be false. If 'Rcc' is true, then '$Rcc \leftrightarrow Rcc$' is true. If 'Rcc' is false, then '$Rcc \leftrightarrow Rcc$' will be true. So either way, '$Rcc \leftrightarrow Rcc$' is true. Since there was nothing special about *obbie*—we might have chosen any object—we see that no matter how we extend our original interpretation by allowing 'c' to name some new object, '$Rcc \leftrightarrow Rcc$' will be true in the new interpretation. So '$\forall x(Rxx \leftrightarrow Rxx)$' was true in the original interpretation. But we chose our interpretation arbitrarily, so '$\forall x(Rxx \leftrightarrow Rxx)$' is true in every interpretation. It is therefore a logical truth.

This is quite longwinded, but, as things stand, there is no alternative. In order to show that a sentence is a logical truth, 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 logically equivalent; for this requires that they have the same truth value in *every* interpretation.
- that some sentences are jointly inconsistent; 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! Let's take just one more example. Here is an argument which is obviously valid:

$$\forall x(Hx \wedge Jx) \therefore \forall xHx$$

After all, if everything is both H and J, then everything is H. But we can only show that the argument is valid by considering what must be true in every interpretation in which the premise is true. To show this, we would have to reason as follows:

> Consider an arbitrary interpretation in which the premise '$\forall x(Hx \wedge Jx)$' is true. It follows that, however we expand the interpretation with a new name, for example 'c', '$Hc \wedge Jc$' will be true in this new interpretation. 'Hc' will, then, also be true in this new interpretation. But since this held for *any* way of expanding the interpretation, it must be that '$\forall xHx$' is true in the old 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 '$\forall xHx$' is true. The argument is valid!

Even for a simple argument like this one, the reasoning is somewhat complicated. For longer arguments, the reasoning can be extremely torturous.

The following table summarises whether a single (counter-)interpretation suffices, or whether we must reason about all interpretations.

	Yes	**No**
logical truth?	all interpretations	one counter-interpretation
contradiction?	all interpretations	one counter-interpretation
equivalent?	all interpretations	one counter-interpretation
consistent?	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 TFL concerns truth tables, whereas FOL 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.

Chapter 7

Natural deduction for FOL

Basic rules for FOL 30

FOL makes use of all of the connectives of TFL. So proofs in FOL will use all of the basic and derived rules from Part 4. 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, and to govern the identity sign.

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:

1	$\forall x Rxxd$	
2	$Raad$	$\forall$E 1

We obtained line 2 by dropping the universal quantifier and replacing every instance of 'x' with 'a'. Equally, the following should be allowed:

1	$\forall x Rxxd$	
2	$Rddd$	$\forall$E 1

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 ($\forall$E):

m	$\forall \boldsymbol{x} \mathscr{A}(\ldots \boldsymbol{x} \ldots \boldsymbol{x} \ldots)$	
	$\mathscr{A}(\ldots \boldsymbol{c} \ldots \boldsymbol{c} \ldots)$	$\forall$E m

The notation here was introduced in §26. The point is that you can obtain any *substitution instance* of a universally quantified formula: replace 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 logical operator. Thus the following is outright banned:

1	$\forall x Bx \to Bk$	
2	$Bb \to Bk$	naughtily attempting to invoke ∀E 1

This is illegitimate, since '$\forall x$' is not the main logical operator in line 1. (If you need a reminder as to why this sort of inference should be banned, reread §22.)

30.2 Existential introduction

From the claim that some particular thing is an F, you can infer that something is an F. So we ought to allow:

1	$Raad$	
2	$\exists x Raax$	∃I 1

Here, we have replaced the name 'd' with a variable 'x', and then existentially quantified over it. Equally, we would have allowed:

1	$Raad$	
2	$\exists x Rxxd$	∃I 1

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. (After all, if Narcissus loves himself, then there is someone who loves Narcissus.) So we would also allow:

1	$Raad$	
2	$\exists x Rxad$	∃I 1

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:

m	$\mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots)$	
	$\exists\mathscr{x}\mathscr{A}(\ldots\mathscr{x}\ldots\mathscr{c}\ldots)$	∃I m

$\mathscr{x}$ must not occur in $\mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots)$

The constraint is included to guarantee that any application of the rule yields a sentence of FOL. Thus the following is allowed:

1	$Raad$	
2	$\exists xRxad$	$\exists$I 1
3	$\exists y\exists xRxyd$	$\exists$I 2

But this is banned:

1	$Raad$	
2	$\exists xRxad$	$\exists$I 1
3	$\exists x\exists xRxxd$	naughtily attempting to invoke $\exists$I 2

since the expression on line 3 contains clashing variables, and so fails to count as a sentence of FOL.

30.3 Empty domains

The following proof combines our two new rules for quantifiers:

1	$\forall xFx$	
2	Fa	$\forall$E 1
3	$\exists xFx$	$\exists$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, '$\exists xFx$' follows from '$\forall xFx$', 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.

Actually, we are already committed to this oddity. In §21, we stipulated that domains in FOL must have at least one member. We then defined a logical truth (of FOL) as a sentence which is true in every interpretation. Since '$\exists x\ x = x$' will be true in every interpretation, this *also* had the effect of stipulating that it is a matter of logic that there is something rather than nothing.

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:

- $\forall xFx \vdash Fa$: after all, that was $\forall$E.
- $Fa \vdash \exists xFx$: after all, that was $\exists$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 $\forall xFx \vdash \exists xFx$. 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 FOL, 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 '$\forall xFx$', then you can infer '$\forall xFx$'.

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 FOL, 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:

$$\forall xFx \therefore \forall yFy$$

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:

1	$\forall xFx$	
2	Fa	$\forall$E 1

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 '$\forall yFy$', thus:

1	$\forall xFx$	
2	Fa	$\forall$E 1
3	$\forall yFy$	$\forall$I 2

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 (∀I):

m	$\mathscr{A}(\ldots \mathscr{c} \ldots \mathscr{c} \ldots)$	
	$\forall x \mathscr{A}(\ldots x \ldots x \ldots)$	∀I m

$\mathscr{c}$ must not occur in any undischarged assumption
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:

1	$\forall x Lxk$	
2	Lkk	∀E 1
3	$\forall x Lxx$	naughtily attempting to invoke ∀I 2

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 as a discharged assumption. That is, it may occur in a subproof that we have already closed. For example:

1	$\quad Gd$	
2	$\quad Gd$	R 1
3	$Gd \to Gd$	→I 1–2
4	$\forall z(Gz \to Gz)$	∀I 3

This tells us that '$\forall z(Gz \to Gz)$' is a *theorem*. And that is as it should be.

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 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 ($\exists$E):

m	$\exists x\mathscr{A}(\ldots x \ldots x \ldots)$	
i	$\quad \mathscr{A}(\ldots c \ldots c \ldots)$	
j	$\quad \mathscr{B}$	
	$\mathscr{B}$	$\exists$E m, i–j

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. There is someone who 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:

1	Lb	
2	$\exists x \neg Lx$	
3	$\quad \neg Lb$	
4	$\quad Lb \wedge \neg Lb$	$\wedge$I 1, 3
5	$Lb \wedge \neg Lb$	naughtily attempting to invoke $\exists$E 2, 3–4

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 following proof would be no better:

1	Lb	
2	$\exists x \neg Lx$	
3	$\quad \neg Lb$	
4	$\quad Lb \wedge \neg Lb$	$\wedge$I 1, 3
5	$\quad \exists x(Lx \wedge \neg Lx)$	$\exists$I 4
6	$\exists x(Lx \wedge \neg Lx)$	naughtily attempting to invoke $\exists$E 2, 3–5

The last line of the proof would 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 $\exists$E.

Practice exercises

A. The following two 'proofs' are *incorrect*. Explain why both are incorrect. Also, provide interpretations which would invalidate the fallacious argument forms the 'proofs' enshrine:

1	$\forall x Rxx$	
2	Raa	$\forall$E 1
3	$\forall y Ray$	$\forall$I 2
4	$\forall x \forall y Rxy$	$\forall$I 3

1	$\forall x \exists y Rxy$	
2	$\exists y Ray$	$\forall$E 1
3	$\quad Raa$	
4	$\quad \exists x Rxx$	$\exists$I 3
5	$\exists x Rxx$	$\exists$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	$\quad Rma \vee Ram$
5	$\quad \neg Rma$
6	$\quad Ram$
7	$\quad \exists x Rxm$
8	$\exists x Rxm$

1	$\forall x (\exists y Lxy \to \forall z Lzx)$
2	Lab
3	$\exists y Lay \to \forall z Lza$
4	$\exists y Lay$
5	$\forall z Lza$
6	Lca
7	$\exists y Lcy \to \forall z Lzc$
8	$\exists y Lcy$
9	$\forall z Lzc$
10	Lcc
11	$\forall x Lxx$

1	$\forall x (Jx \to Kx)$
2	$\exists x \forall y Lxy$
3	$\forall x Jx$
4	$\quad \forall y Lay$
5	$\quad Laa$
6	$\quad Ja$
7	$\quad Ja \to Ka$
8	$\quad Ka$
9	$\quad Ka \wedge Laa$
10	$\quad \exists x (Kx \wedge Lxx)$
11	$\exists x (Kx \wedge Lxx)$

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 the following argument forms in FOL 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. $\forall x(Mx \leftrightarrow Nx), Ma \land \exists x Rxa \vdash \exists x Nx$
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. $\forall x \forall y(Gxy \to Gyx) \vdash \forall x \forall y(Gxy \leftrightarrow Gyx)$
10. $\forall x(\neg Mx \lor Ljx), \forall x(Bx \to Ljx), \forall x(Mx \lor Bx) \vdash \forall x Ljx$

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. $\forall x(Ax \to \neg Bx)$, $\neg \exists x(Ax \land Bx)$
2. $\forall x(\neg Ax \to Bd)$, $\forall x Ax \lor Bd$
3. $\exists x Px \to Qc$, $\forall x(Px \to Qc)$

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 logically equivalent.

1. $\forall x Px \to Qc, \forall x(Px \to Qc)$
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 FOL, 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. $\exists x Mx \rightarrow \exists x Nx, \neg \exists x Nx \therefore \forall x \neg Mx$

Conversion of quantifiers 31

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:

m	$\forall x\neg\mathscr{A}$	
	$\neg\exists x\mathscr{A}$	CQ m

and

m	$\neg\exists x\mathscr{A}$	
	$\forall x\neg\mathscr{A}$	CQ m

Equally, we add:

m	$\exists x\neg\mathscr{A}$	
	$\neg\forall x\mathscr{A}$	CQ m

and

m	$\neg\forall x\mathscr{A}$	
	$\exists x\neg\mathscr{A}$	CQ m

Practice exercises

A. Show in each case that the sentences are provably inconsistent:

1. $Sa \to Tm, Tm \to Sa, Tm \wedge \neg Sa$
2. $\neg\exists xRxa, \forall x\forall yRyx$
3. $\neg\exists x\exists yLxy, Laa$
4. $\forall x(Px \to Qx), \forall z(Pz \to Rz), \forall yPy, \neg Qa \wedge \neg Rb$

B. Show that each pair of sentences is provably equivalent:

1. $\forall x(Ax \to \neg Bx), \neg\exists x(Ax \wedge Bx)$
2. $\forall x(\neg Ax \to Bd), \forall x Ax \vee Bd$

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. $\forall x(Fx \wedge Ga), \forall x Fx \wedge Ga$
2. $\exists x(Fx \vee Ga), \exists x Fx \vee Ga$
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.

Derived rules 32

As in the case of TFL, we first introduced some rules for FOL as basic (in §30), and then added some further rules for conversion of quantifiers (in §31). 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	$\forall x \neg Ax$	
2	$\quad \exists x Ax$	
3	$\quad\quad Ac$	
4	$\quad\quad \neg Ac$	$\forall$E 1
5	$\quad\quad \bot$	$\bot$I 3, 4
6	$\quad \bot$	$\exists$E 2, 3–5
7	$\neg \exists x Ax$	$\neg$I 2–6

Here is a justification of the third CQ rule:

1	$\exists x \neg Ax$	
2	$\quad \forall x Ax$	
3	$\quad\quad \neg Ac$	
4	$\quad\quad Ac$	$\forall$E 2
5	$\quad\quad \bot$	$\bot$I 4, 3
6	$\quad \bot$	$\exists$E 1, 3–5
7	$\neg \forall x Ax$	$\neg$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.

Proof-theoretic and semantic concepts 33

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 there is no valuation (or interpretation) which makes all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ true and makes $\mathscr{C}$ 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 logical truths 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 logical truth, though, requires reasoning about all possible interpretations. Given a choice between showing that a sentence is a theorem and showing that it is a logical truth, 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 logical truth, 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 logical truth, it would be easier to show that it is not a logical truth.

Fortunately, *a sentence is a theorem if and only if it is a logical truth.* As a result, if we provide a proof of $\mathscr{A}$ on no assumptions, and thus show that $\mathscr{A}$ is a theorem, we can legitimately infer that $\mathscr{A}$ is a logical truth; 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 logical truth, it follows that $\mathscr{A}$ is not a theorem.

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 *provably consistent* 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 **logical truth**?	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 $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ **jointly consistent**?	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

Chapter 8

Identity

Identity 34

Consider this sentence:

1. Pavel owes money to everyone

Let the domain be people; this will allow us to symbolize 'everyone' as a universal quantifier. Offering the symbolization key:

Oxy: $_____x$ owes money to $_____y$
p: Pavel

we can symbolize sentence 1 by '$\forall xOpx$'. But this has a (perhaps) odd consequence. It requires that Pavel owes money to every member of the domain (whatever the domain may be). The domain certainly includes Pavel. So this entails that Pavel owes money to himself.

Perhaps we meant to say:

2. Pavel owes money to everyone *else*
3. Pavel owes money to everyone *other than* Pavel
4. Pavel owes money to everyone *except* Pavel himself

but we do not know how to deal with the italicised words yet. The solution is to add another symbol to FOL.

34.1 Adding identity

The symbol '=' is a two-place predicate. Since it is to have a special meaning, we will write it a bit differently: we put it between two terms, rather than out front. And it *does* have a very particular meaning. We *always* adopt the following symbolization key:

$x = y$: $_____x$ is identical to $_____y$

This does not mean *merely* that the objects in question are indistinguishable, or that all of the same things are true of them. Rather, it means that the objects in question are *the very same* object.

Now suppose we want to symbolize this sentence:

5. Pavel is Mister Checkov.

Let us add to our symbolization key:

c: Mister Checkov

Now sentence 5 can be symbolized as '$p = c$'. This means that the names 'p' and 'c' both name the same thing.

We can also now deal with sentences 2–4. All of these sentences can be paraphrased as 'Everyone who is not Pavel is owed money by Pavel'. Paraphrasing some more, we get: 'For all x, if x is not Pavel, then x is owed money by Pavel'. Now that we are armed with our new identity symbol, we can symbolize this as '$\forall x(\neg x = p \rightarrow Opx)$'.

This last sentence contains the formula '$\neg x = p$'. That might look a bit strange, because the symbol that comes immediately after the '$\neg$' is a variable, rather than a predicate, but this is not a problem. We are simply negating the entire formula, '$x = p$'.

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

6. No one besides Pavel owes money to Hikaru.
7. Only Pavel owes Hikaru money.

Let 'h' name Hikaru. Sentence 6 can be paraphrased as, 'No one who is not Pavel owes money to Hikaru'. This can be symbolized by '$\neg\exists x(\neg x = p \wedge Oxh)$'. Equally, sentence 6 can be paraphrased as 'for all x, if x owes money to Hikaru, then x is Pavel'. It can then be symbolized as '$\forall x(Oxh \rightarrow x = p)$'.

Sentence 7 can be treated similarly, but there is one subtlety here. Do either sentence 6 or 7 entail that Pavel himself owes money to Hikaru?

34.2 There are at least...

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

8. There is at least one apple
9. There are at least two apples
10. There are at least three apples

We will use the symbolization key:

Ax: $_____{}_x$ is an apple

Sentence 8 does not require identity. It can be adequately symbolized by '$\exists x Ax$': There is an apple; perhaps many, but at least one.

It might be tempting to also symbolize sentence 9 without identity. Yet consider the sentence '$\exists x \exists y(Ax \wedge Ay)$'. Roughly, this says that there is some apple x in the domain and some apple y in the domain. Since nothing precludes these from being one and the same apple, this would be true even if there were only one apple. In order to make sure that we are dealing with *different* apples, we need an identity predicate. Sentence 9 needs to say that the two apples that exist are not identical, so it can be symbolized by '$\exists x \exists y((Ax \wedge Ay) \wedge \neg x = y)$'.

Sentence 10 requires talking about three different apples. Now we need three existential quantifiers, and we need to make sure that each will pick out something different: '$\exists x \exists y \exists z[((Ax \wedge Ay) \wedge Az) \wedge ((\neg x = y \wedge \neg y = z) \wedge \neg x = z)]$'.

34.3 There are at most...

Now consider these sentences:

11. There is at most one apple
12. There are at most two apples

Sentence 11 can be paraphrased as, 'It is not the case that there are at least *two* apples'. This is just the negation of sentence 9:

$$\neg\exists x\exists y[(Ax \wedge Ay) \wedge \neg x = y]$$

But sentence 11 can also be approached in another way. It means that if you pick out an object and it's an apple, and then you pick out an object and it's also an apple, you must have picked out the same object both times. With this in mind, it can be symbolized by

$$\forall x\forall y\big[(Ax \wedge Ay) \rightarrow x = y\big]$$

The two sentences will turn out to be logically equivalent.

In a similar way, sentence 12 can be approached in two equivalent ways. It can be paraphrased as, 'It is not the case that there are *three* or more distinct apples', so we can offer:

$$\neg\exists x\exists y\exists z(Ax \wedge Ay \wedge Az \wedge \neg x = y \wedge \neg y = z \wedge \neg x = z)$$

Alternatively we can read it as saying that if you pick out an apple, and an apple, and an apple, then you will have picked out (at least) one of these objects more than once. Thus:

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

34.4 There are exactly...

We can now consider precise statements, like:

13. There is exactly one apple.
14. There are exactly two apples.
15. There are exactly three apples.

Sentence 13 can be paraphrased as, 'There is *at least* one apple and there is *at most* one apple'. This is just the conjunction of sentence 8 and sentence 11. So we can offer:

$$\exists x Ax \wedge \forall x\forall y\big[(Ax \wedge Ay) \rightarrow x = y\big]$$

But it is perhaps more straightforward to paraphrase sentence 13 as, 'There is a thing x which is an apple, and everything which is an apple is just x itself'. Thought of in this way, we offer:

$$\exists x\big[Ax \wedge \forall y(Ay \rightarrow x = y)\big]$$

Similarly, sentence 14 may be paraphrased as, 'There are *at least* two apples, and there are *at most* two apples'. Thus we could offer

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

More efficiently, though, we can paraphrase it as 'There are at least two different apples, and every apple is one of those two apples'. Then we offer:

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

Finally, consider these sentence:

16. There are exactly two things
17. There are exactly two objects

It might be tempting to add a predicate to our symbolization key, to symbolize the English predicate '_____ is a thing' or '_____ is an object', but this is unnecessary. Words like 'thing' and 'object' do not sort wheat from chaff: they apply trivially to everything, which is to say, they apply trivially to every thing. So we can symbolize either sentence with either of the following:

$$\exists x \exists y \neg x = y \wedge \neg \exists x \exists y \exists z ((\neg x = y \wedge \neg y = z) \wedge \neg x = z)$$
$$\exists x \exists y \big[\neg x = y \wedge \forall z (x = z \vee y = z)\big]$$

Practice exercises

A. Explain why:

- '$\exists x \forall y (Ay \leftrightarrow x = y)$' is a good symbolization of 'there is exactly one apple'.
- '$\exists x \exists y \big[\neg x = y \wedge \forall z (Az \leftrightarrow (x = z \vee y = z))\big]$' is a good symbolization of 'there are exactly two apples'.

Definite descriptions 35

Consider sentences like:

1. Nick is the traitor.
2. The traitor went to Cambridge.
3. The traitor is the deputy

These are definite descriptions: they are meant to pick out a *unique* object. They should be contrasted with *indefinite* descriptions, such as 'Nick is *a* traitor'. They should equally be contrasted with *generics*, such as '*The* whale is a mammal' (it's inappropriate to ask *which* whale). The question we face is: how should we deal with definite descriptions in FOL?

35.1 Treating definite descriptions as terms

One option would be to introduce new names whenever we come across a definite description. This is probably not a great idea. We know that *the* traitor—whoever it is—is indeed *a* traitor. We want to preserve that information in our symbolization.

A second option would be to use a *new* definite description operator, such as '$\imath$'. The idea would be to symbolize 'the F' as '$\imath x Fx$'; or to symbolize 'the G' as '$\imath x Gx$', etc. Expression of the form $\imath x \mathcal{A}x$ would then behave like names. If we followed this path, then using the following symbolization key:

domain: people
Tx: $______x$ is a traitor
Dx: $______x$ is a deputy
Cx: $______x$ went to Cambridge
n: Nick

We could symbolize sentence 1 with '$\imath x Tx = n$', sentence 2 with '$C\imath x Tx$', and sentence 3 with '$\imath x Tx = \imath x Dx$'.

However, it would be nice if we didn't have to add a new symbol to FOL. And indeed, we might be able to make do without one.

35.2 Russell's analysis

Bertrand Russell offered an analysis of definite descriptions. Very briefly put, he observed that, when we say 'the F' in the context of a definite description,

our aim is to pick out the *one and only* thing that is F (in the appropriate context). Thus Russell analysed the notion of a definite description as follows:[1]

the F is G **iff** there is at least one F, *and*
there is at most one F, *and*
every F is G

Note a very important feature of this analysis: *'the' does not appear on the right-side of the equivalence.* Russell is aiming to provide an understanding of definite descriptions in terms that do not presuppose them.

Now, one might worry that we can say 'the table is brown' without implying that there is one and only one table in the universe. But this is not (yet) a fantastic counterexample to Russell's analysis. The domain of discourse is likely to be restricted by context (e.g. to objects in my line of sight).

If we accept Russell's analysis of definite descriptions, then we can symbolize sentences of the form 'the F is G' using our strategy for numerical quantification in FOL. After all, we can deal with the three conjuncts on the right-hand side of Russell's analysis as follows:

$$\exists x Fx \wedge \forall x \forall y((Fx \wedge Fy) \rightarrow x = y) \wedge \forall x(Fx \rightarrow Gx)$$

In fact, we could express the same point rather more crisply, by recognizing that the first two conjuncts just amount to the claim that there is *exactly* one F, and that the last conjunct tells us that that object is F. So, equivalently, we could offer:

$$\exists x\big[(Fx \wedge \forall y(Fy \rightarrow x = y)) \wedge Gx\big]$$

Using these sorts of techniques, we can now symbolize sentences 1–3 without using any new-fangled fancy operator, such as '$\imath$'.

Sentence 1 is exactly like the examples we have just considered. So we would symbolize it by '$\exists x(Tx \wedge \forall y(Ty \rightarrow x = y) \wedge x = n)$'.

Sentence 2 poses no problems either: '$\exists x(Tx \wedge \forall y(Ty \rightarrow x = y) \wedge Cx)$'.

Sentence 3 is a little trickier, because it links two definite descriptions. But, deploying Russell's analysis, it can be paraphrased by 'there is exactly one traitor, x, and there is exactly one deputy, y, and x = y'. So we can symbolize it by:

$$\exists x \exists y\big(\big[Tx \wedge \forall z(Tz \rightarrow x = z)\big] \wedge \big[Dy \wedge \forall z(Dz \rightarrow y = z)\big] \wedge x = y\big)$$

Note that we have made sure that the formula '$x = y$' falls within the scope of both quantifiers!

35.3 Empty definite descriptions

One of the nice features of Russell's analysis is that it allows us to handle *empty* definite descriptions neatly.

France has no king at present. Now, if we were to introduce a name, 'k', to name the present King of France, then everything would go wrong: remember

[1] Bertrand Russell, 'On Denoting', 1905, *Mind 14*, pp. 479–93; also Russell, *Introduction to Mathematical Philosophy*, 1919, London: Allen and Unwin, ch. 16.

from §21 that a name must always pick out some object in the domain, and whatever we choose as our domain, it will contain no present kings of France.

Russell's analysis neatly avoids this problem. Russell tells us to treat definite descriptions using predicates and quantifiers, instead of names. Since predicates can be empty (see §22), this means that no difficulty now arises when the definite description is empty.

Indeed, Russell's analysis helpfully highlights two ways to go wrong in a claim involving a definite description. To adapt an example from Stephen Neale (1990),[2] suppose Alex claims:

4. I am dating the present king of France.

Using the following symbolization key:

a: Alex
Kx: $______x$ is a present king of France
Dxy: $______x$ is dating $______y$

Sentence 4 would be symbolized by '$\exists x(\forall y(Ky \leftrightarrow x = y) \wedge Dax)$'. Now, this can be false in (at least) two ways, corresponding to these two different sentences:

5. There is no one who is both the present King of France and such that he and Alex are dating.
6. There is a unique present King of France, but Alex is not dating him.

Sentence 5 might be paraphrased by 'It is not the case that: the present King of France and Alex are dating'. It will then be symbolized by '$\neg\exists x\big[(Kx \wedge \forall y(Ky \rightarrow x = y)) \wedge Dax\big]$'. We might call this *outer* negation, since the negation governs the entire sentence. Note that it will be true if there is no present King of France.

Sentence 6 can be symbolized by '$\exists x((Kx \wedge \forall y(Ky \rightarrow x = y)) \wedge \neg Dax)$'. We might call this *inner* negation, since the negation occurs within the scope of the definite description. Note that its truth requires that there is a present King of France, albeit one who is not dating Alex.

35.4 The adequacy of Russell's analysis

How good is Russell's analysis of definite descriptions? This question has generated a substantial philosophical literature, but we will restrict ourselves to two observations.

One worry focusses on Russell's treatment of empty definite descriptions. If there are no Fs, then on Russell's analysis, both 'the F is G' is and 'the F is non-G' are false. P.F. Strawson suggested that such sentences should not be regarded as false, exactly.[3] Rather, they involve presupposition failure, and need to be regarded as *neither* true *nor* false.

If we agree with Strawson here, we will need to revise our logic. For, in our logic, there are only two truth values (True and False), and every sentence is assigned exactly one of these truth values.

[2] Neale, *Descriptions*, 1990, Cambridge: MIT Press.
[3] P.F. Strawson, 'On Referring', 1950, *Mind 59*, pp. 320–34.

But there is room to disagree with Strawson. Strawson is appealing to some linguistic intuitions, but it is not clear that they are very robust. For example: isn't it just *false*, not 'gappy', that Tim is dating the present King of France?

Keith Donnellan raised a second sort of worry, which (very roughly) can be brought out by thinking about a case of mistaken identity.[4] Two men stand in the corner: a very tall man drinking what looks like a gin martini; and a very short man drinking what looks like a pint of water. Seeing them, Malika says:

7. The gin-drinker is very tall!

Russell's analysis will have us render Malika's sentence as:

7′. There is exactly one gin-drinker [in the corner], and whoever is a gin-drinker [in the corner] is very tall.

Now suppose that the very tall man is actually drinking *water* from a martini glass; whereas the very short man is drinking a pint of (neat) gin. By Russell's analysis, Malika has said something false, but don't we want to say that Malika has said something *true*?

Again, one might wonder how clear our intuitions are on this case. We can all agree that Malika intended to pick out a particular man, and say something true of him (that he was tall). On Russell's analysis, she actually picked out a different man (the short one), and consequently said something false of him. But maybe advocates of Russell's analysis only need to explain *why* Malika's intentions were frustrated, and so why she said something false. This is easy enough to do: Malika said something false because she had false beliefs about the men's drinks; if Malika's beliefs about the drinks had been true, then she would have said something true.[5]

To say much more here would lead us into deep philosophical waters. That would be no bad thing, but for now it would distract us from the immediate purpose of learning formal logic. So, for now, we will stick with Russell's analysis of definite descriptions, when it comes to putting things into FOL. It is certainly the best that we can offer, without significantly revising our logic, and it is quite defensible as an analysis.

Practice exercises

A. Using the following symbolization key:

domain: people
Kx: _____$_x$ knows the combination to the safe.
Sx: _____$_x$ is a spy.
Vx: _____$_x$ is a vegetarian.
Txy: _____$_x$ trusts _____$_y$.
h: Hofthor
i: Ingmar

[4] Keith Donnellan, 'Reference and Definite Descriptions', 1966, *Philosophical Review 77*, pp. 281–304.

[5] Interested parties should read Saul Kripke, 'Speaker Reference and Semantic Reference', 1977, in French et al (eds.), *Contemporary Perspectives in the Philosophy of Language*, Minneapolis: University of Minnesota Press, pp. 6-27.

symbolize the following sentences in FOL:

1. Hofthor trusts a vegetarian.
2. Everyone who trusts Ingmar trusts a vegetarian.
3. Everyone who trusts Ingmar trusts someone who trusts a vegetarian.
4. Only Ingmar knows the combination to the safe.
5. Ingmar trusts Hofthor, but no one else.
6. The person who knows the combination to the safe is a vegetarian.
7. The person who knows the combination to the safe is not a spy.

B. Using the following symbolization key:

domain: cards in a standard deck
Bx: $_____{}_x$ is black.
Cx: $_____{}_x$ is a club.
Dx: $_____{}_x$ is a deuce.
Jx: $_____{}_x$ is a jack.
Mx: $_____{}_x$ is a man with an axe.
Ox: $_____{}_x$ is one-eyed.
Wx: $_____{}_x$ is wild.

symbolize each sentence in FOL:

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

C. Using the following symbolization key:

domain: animals in the world
Bx: $_____{}_x$ is in Farmer Brown's field.
Hx: $_____{}_x$ is a horse.
Px: $_____{}_x$ is a Pegasus.
Wx: $_____{}_x$ has wings.

symbolize the following sentences in FOL:

1. There are at least three horses in the world.
2. There are at least three animals in the world.
3. There is more than one horse in Farmer Brown's field.
4. There are three horses in Farmer Brown's field.
5. There is a single winged creature in Farmer Brown's field; any other creatures in the field must be wingless.

6. The Pegasus is a winged horse.
7. The animal in Farmer Brown's field is not a horse.
8. The horse in Farmer Brown's field does not have wings.

D. In this chapter, we symbolized 'Nick is the traitor' by '$\exists x(Tx \wedge \forall y(Ty \rightarrow x = y) \wedge x = n)$'. Two equally good symbolizations would be:

- $Tn \wedge \forall y(Ty \rightarrow n = y)$
- $\forall y(Ty \leftrightarrow y = n)$

Explain why these would be equally good symbolizations.

Semantics for FOL with identity 36

We first need to expand our notion of what it is to be a sentence

36.1 Sentences of FOL with identity

When we add the identity symbol to FOL, we add a new kind of atomic sentence, for example $a = b$.

We simply do this by adding a new kind of atomic sentence:

> If $\mathscr{a}$ and $\mathscr{b}$ are names, then $\mathscr{a} = \mathscr{b}$ is an atomic sentence.

36.2 Semantics for identity

Identity is a special predicate of FOL. We write it a bit differently than other two-place predicates: '$x = y$' instead of 'Ixy' (for example). More important, though, its interpretation is fixed, once and for all.

If two names refer to the same object, then swapping one name for another will not change the truth value of any sentence. So, in particular, if 'a' and 'b' name the same object, then all of the following will be true:

$$Aa \leftrightarrow Ab$$
$$Ba \leftrightarrow Bb$$
$$Raa \leftrightarrow Rbb$$
$$Raa \leftrightarrow Rab$$
$$Rca \leftrightarrow Rcb$$
$$\forall x Rxa \leftrightarrow \forall x Rxb$$

Some philosophers have believed the reverse of this claim. That is, they have believed that when exactly the same sentences (not containing '=') are true of two objects, then they are really just one and the same object after all. This is a highly controversial philosophical claim (sometimes called the *identity of indiscernibles*) and our logic will not subscribe to it; we allow that exactly the same things might be true of two *distinct* objects.

To bring this out, consider the following interpretation:

domain: P.D. Magnus, Tim Button
  a: P.D. Magnus
  b: Tim Button

- For every primitive predicate we care to consider, that predicate is true of *nothing*.

Suppose 'A' is a one-place predicate; then 'Aa' is false and 'Ab' is false, so '$Aa \leftrightarrow Ab$' is true. Similarly, if 'R' is a two-place predicate, then 'Raa' is false and 'Rab' is false, so that '$Raa \leftrightarrow Rab$' is true. And so it goes: every atomic sentence not involving '=' is false, so every biconditional linking such sentences is true. For all that, Tim Button and P.D. Magnus are two distinct people, not one and the same!

Where $\mathscr{a}$ and $\mathscr{b}$ are any names,

> $\mathscr{a} = \mathscr{b}$ is true in an interpretation **iff**
> $\mathscr{a}$ and $\mathscr{b}$ name the very same object in that interpretation

So in our go-to interpretation, '$a = b$' is false, since Aristotle is distinct from Beyoncé.

Rules for identity 37

In §25, we mentioned the philosophically contentious thesis of the *identity of indiscernibles*. This is the claim that objects which are indiscernible in every way are, in fact, identical to each other. It was also mentioned that we will not subscribe to this thesis. It follows that, no matter how much you learn about two objects, we cannot prove that they are identical. That is unless, of course, you learn that the two objects are, in fact, identical, but then the proof will hardly be very illuminating.

The consequence of this, for our proof system, is that there are no sentences that do not already contain the identity predicate that could justify the conclusion '$a = b$'. This means that the identity introduction rule will not justify '$a = b$', or any other identity claim containing two different names.

However, every object is identical to itself. No premises, then, are required in order to conclude that something is identical to itself. So this will be the identity introduction rule:

	$\mathscr{c} = \mathscr{c}$	=I

Notice that this rule does not require referring to any prior lines of the proof. For any name $\mathscr{c}$, you can write $\mathscr{c} = \mathscr{c}$ on any point, with only the =I rule as justification.

Our elimination rule is more fun. If you have established '$a = b$', then anything that is true of the object named by 'a' must also be true of the object named by 'b'. For any sentence with 'a' in it, you can replace some or all of the occurrences of 'a' with 'b' and produce an equivalent sentence. For example, from 'Raa' and '$a = b$', you are justified in inferring 'Rab', 'Rba' or 'Rbb'. More generally:

m	$\mathscr{a} = \mathscr{b}$	
n	$\mathscr{A}(\ldots\mathscr{a}\ldots\mathscr{a}\ldots)$	
	$\mathscr{A}(\ldots\mathscr{b}\ldots\mathscr{a}\ldots)$	=E m, n

The notation here is as for ∃I. So $\mathscr{A}(\ldots\mathscr{a}\ldots\mathscr{a}\ldots)$ is a formula containing the name $\mathscr{a}$, and $\mathscr{A}(\ldots\mathscr{b}\ldots\mathscr{a}\ldots)$ is a formula obtained by replacing one or more instances of the name $\mathscr{a}$ with the name $\mathscr{b}$. Lines m and n can occur in either order, and do not need to be adjacent, but we always cite the statement

of identity first. Symmetrically, we allow:

m	$\mathscr{a} = \mathscr{b}$	
n	$\mathscr{A}(\ldots \mathscr{b} \ldots \mathscr{b} \ldots)$	
	$\mathscr{A}(\ldots \mathscr{a} \ldots \mathscr{b} \ldots)$	=E m, n

This rule is sometimes called *Leibniz's Law*, after Gottfried Leibniz.

To see the rules in action, we will prove some quick results. First, we will prove that identity is *symmetric*:

1	$\quad a = b$	
2	$\quad a = a$	=I
3	$\quad b = a$	=E 1, 2
4	$a = b \to b = a$	→I 1–3
5	$\forall y(a = y \to y = a)$	∀I 4
6	$\forall x \forall y(x = y \to y = x)$	∀I 5

We obtain line 3 by replacing one instance of 'a' in line 2 with an instance of 'b'; this is justified given '$a = b$'.

Second, we will prove that identity is *transitive*:

1	$\quad a = b \land b = c$	
2	$\quad a = b$	∧E 1
3	$\quad b = c$	∧E 1
4	$\quad a = c$	=E 2, 3
5	$(a = b \land b = c) \to a = c$	→I 1–4
6	$\forall z((a = b \land b = z) \to a = z)$	∀I 5
7	$\forall y \forall z((a = y \land y = z) \to a = z)$	∀I 6
8	$\forall x \forall y \forall z((x = y \land y = z) \to x = z)$	∀I 7

We obtain line 4 by replacing 'b' in line 3 with 'a'; this is justified given '$a = b$'.

Practice exercises

A. Provide a proof of each claim.

1. $Pa \lor Qb, Qb \to b = c, \neg Pa \vdash Qc$
2. $m = n \lor n = o, An \vdash Am \lor Ao$
3. $\forall x\ x = m, Rma \vdash \exists x Rxx$
4. $\forall x \forall y(Rxy \to x = y) \vdash Rab \to Rba$

5. $\neg\exists x\neg x = m \vdash \forall x\forall y(Px \to Py)$
6. $\exists xJx, \exists x\neg Jx \vdash \exists x\exists y\ \neg x = y$
7. $\forall x(x = n \leftrightarrow Mx), \forall x(Ox \vee \neg Mx) \vdash On$
8. $\exists xDx, \forall x(x = p \leftrightarrow Dx) \vdash Dp$
9. $\exists x\big[(Kx \wedge \forall y(Ky \to x = y)) \wedge Bx\big], Kd \vdash Bd$
10. $\vdash Pa \to \forall x(Px \vee \neg x = a)$

B. Show that the following are provably equivalent:

- $\exists x\big([Fx \wedge \forall y(Fy \to x = y)] \wedge x = n\big)$
- $Fn \wedge \forall y(Fy \to n = y)$

And hence that both have a decent claim to symbolize the English sentence 'Nick is the F'.

C. In §34, we claimed that the following are logically equivalent symbolizations of the English sentence 'there is exactly one F':

- $\exists xFx \wedge \forall x\forall y\big[(Fx \wedge Fy) \to x = y\big]$
- $\exists x\big[Fx \wedge \forall y(Fy \to x = y)\big]$
- $\exists x\forall y(Fy \leftrightarrow x = y)$

Show that they are all provably equivalent. (*Hint*: to show that three claims are provably equivalent, it suffices to show that the first proves the second, the second proves the third and the third proves the first; think about why.)

D. Symbolize the following argument

> There is exactly one F. There is exactly one G. Nothing is both F and G. So: there are exactly two things that are either F or G.

And offer a proof of it.

Chapter 9

Advanced Topics (Non-examinable)

Normal forms and expressive completeness 38

38.1 Disjunctive Normal Form

Sometimes it is useful to consider sentences of a particularly simple form. For instance, we might consider sentences in which $\neg$ only attaches to atomic sentences, or those which are combinations of atomic sentences and negated atomic sentences using only $\wedge$. A relatively general but still simple form is that where a sentence is a disjunction of conjunctions of atomic or negated atomic sentences. When such a sentence is constructed, we start with atomic sentences, then (perhaps) attach negations, then (perhaps) combine using $\wedge$, and finally (perhaps) combine using $\vee$.

Let's say that a sentence is in DISJUNCTIVE NORMAL FORM *iff* it meets all of the following conditions:

(DNF1) No connectives occur in the sentence other than negations, conjunctions and disjunctions;
(DNF2) Every occurrence of negation has minimal scope (i.e. any '$\neg$' is immediately followed by an atomic sentence);
(DNF3) No disjunction occurs within the scope of any conjunction.

So, here are are some sentences in disjunctive normal form:

$$A$$
$$(A \wedge \neg B \wedge C)$$
$$(A \wedge B) \vee (A \wedge \neg B)$$
$$(A \wedge B) \vee (A \wedge B \wedge C \wedge \neg D \wedge \neg E)$$
$$A \vee (C \wedge \neg P_{234} \wedge P_{233} \wedge Q) \vee \neg B$$

Note that we have here broken one of the maxims of this book and *temporarily* allowed ourselves to employ the relaxed bracketing-conventions that allow conjunctions and disjunctions to be of arbitrary length. These conventions make it easier to see when a sentence is in disjunctive normal form. We will continue to help ourselves to these relaxed conventions, without further comment.

To further illustrate the idea of disjunctive normal form, we will introduce some more notation. We write '$\pm\mathscr{A}$' to indicate that $\mathscr{A}$ is an atomic sentence which may or may not be prefaced with an occurrence of negation. Then a sentence in disjunctive normal form has the following shape:

$$(\pm\mathscr{A}_1 \wedge \ldots \wedge \pm\mathscr{A}_i) \vee (\pm\mathscr{A}_{i+1} \wedge \ldots \wedge \pm\mathscr{A}_j) \vee \ldots \vee (\pm\mathscr{A}_{m+1} \wedge \ldots \wedge \pm\mathscr{A}_n)$$

We now know what it is for a sentence to be in disjunctive normal form. The result that we are aiming at is:

Disjunctive Normal Form Theorem. For any sentence, there is a logically equivalent sentence in disjunctive normal form.

Henceforth, we will abbreviate 'Disjunctive Normal Form' by 'DNF'.

38.2 Proof of DNF Theorem via truth tables

Our first proof of the DNF Theorem employs truth tables. We will first illustrate the technique for finding an equivalent sentence in DNF, and then turn this illustration into a rigorous proof.

Let's suppose we have some sentence, $\mathcal{S}$, which contains three atomic sentences, 'A', 'B' and 'C'. The very first thing to do is fill out a complete truth table for $\mathcal{S}$. Maybe we end up with this:

A	B	C	$\mathcal{S}$
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	T

As it happens, $\mathcal{S}$ is true on four lines of its truth table, namely lines 1, 3, 7 and 8. Corresponding to each of those lines, we will write down four sentences, whose only connectives are negations and conjunctions, where every negation has minimal scope:

- '$A \wedge B \wedge C$' which is true on line 1 (and only then)
- '$A \wedge \neg B \wedge C$' which is true on line 3 (and only then)
- '$\neg A \wedge \neg B \wedge C$' which is true on line 7 (and only then)
- '$\neg A \wedge \neg B \wedge \neg C$' which is true on line 8 (and only then)

We now combine all of these conjunctions using $\vee$, like so:

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

This gives us a sentence in DNF which is true on exactly those lines where one of the disjuncts is true, i.e. it is true on (and only on) lines 1, 3, 7, and 8. So this sentence has exactly the same truth table as $\mathcal{S}$. So we have a sentence in DNF that is logically equivalent to $\mathcal{S}$, which is exactly what we wanted!

Now, the strategy that we just adopted did not depend on the specifics of $\mathcal{S}$; it is perfectly general. Consequently, we can use it to obtain a simple proof of the DNF Theorem.

Pick any arbitrary sentence, $\mathcal{S}$, and let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be the atomic sentences that occur in $\mathcal{S}$. To obtain a sentence in DNF that is logically equivalent $\mathcal{S}$, we consider $\mathcal{S}$'s truth table. There are two cases to consider:

1. *$\mathcal{S}$ is false on every line of its truth table.* Then, $\mathcal{S}$ is a contradiction. In that case, the contradiction $(\mathcal{A}_1 \wedge \neg\mathcal{A}_1)$ is in DNF and logically equivalent to $\mathcal{S}$.

2. *$\mathcal{S}$ is true on at least one line of its truth table.* For each line i of the truth table, let $\mathcal{B}_i$ be a conjunction of the form

$$(\pm\mathcal{A}_1 \wedge \ldots \wedge \pm\mathcal{A}_n)$$

where the following rules determine whether or not to include a negation in front of each atomic sentence:

$$\mathcal{A}_m \text{ is a conjunct of } \mathcal{B}_i \textit{ iff } \mathcal{A}_m \text{ is true on line } i$$
$$\neg\mathcal{A}_m \text{ is a conjunct of } \mathcal{B}_i \textit{ iff } \mathcal{A}_m \text{ is false on line } i$$

Given these rules, $\mathcal{B}_i$ is true on (and only on) line i of the truth table which considers all possible valuations of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ (i.e. $\mathcal{S}$'s truth table).

Next, let i_1, i_2, ..., i_m be the numbers of the lines of the truth table where $\mathcal{S}$ is *true*. Now let $\mathcal{D}$ be the sentence:

$$\mathcal{B}_{i_1} \vee \mathcal{B}_{i_2} \vee \ldots \vee \mathcal{B}_{i_m}$$

Since $\mathcal{S}$ is true on at least one line of its truth table, $\mathcal{D}$ is indeed well-defined; and in the limiting case where $\mathcal{S}$ is true on exactly one line of its truth table, $\mathcal{D}$ is just $\mathcal{B}_{i_1}$, for some i_1.

By construction, $\mathcal{D}$ is in DNF. Moreover, by construction, for each line i of the truth table: $\mathcal{S}$ is true on line i of the truth table *iff* one of $\mathcal{D}$'s disjuncts (namely, $\mathcal{B}_i$) is true on, and only on, line i. Hence $\mathcal{S}$ and $\mathcal{D}$ have the same truth table, and so are logically equivalent.

These two cases are exhaustive and, either way, we have a sentence in DNF that is logically equivalent to $\mathcal{S}$.

So we have proved the DNF Theorem. Before we say any more, though, we should immediately flag that we are hereby returning to the austere definition of a (TFL) sentence, according to which we can assume that any conjunction has exactly two conjuncts, and any disjunction has exactly two disjuncts.

38.3 Conjunctive Normal Form

So far in this chapter, we have discussed *disjunctive* normal form. It may not come as a surprise to hear that there is also such a thing as *conjunctive normal form* (CNF).

The definition of CNF is exactly analogous to the definition of DNF. So, a sentence is in CNF *iff* it meets all of the following conditions:

(CNF1) No connectives occur in the sentence other than negations, conjunctions and disjunctions;
(CNF2) Every occurrence of negation has minimal scope;
(CNF3) No conjunction occurs within the scope of any disjunction.

Generally, then, a sentence in CNF looks like this

$$(\pm\mathscr{A}_1 \vee \ldots \vee \pm\mathscr{A}_i) \wedge (\pm\mathscr{A}_{i+1} \vee \ldots \vee \pm\mathscr{A}_j) \wedge \ldots \wedge (\pm\mathscr{A}_{m+1} \vee \ldots \vee \pm\mathscr{A}_n)$$

where each $\mathscr{A}_k$ is an atomic sentence.

We can now prove another normal form theorem:

Conjunctive Normal Form Theorem. For any sentence, there is a logically equivalent sentence in conjunctive normal form.

Given a TFL sentence, $\mathscr{S}$, we begin by writing down the complete truth table for $\mathscr{S}$.

If $\mathscr{S}$ is *true* on every line of the truth table, then $\mathscr{S}$ and $(\mathscr{A}_1 \vee \neg\mathscr{A}_1)$ are logically equivalent.

If $\mathscr{S}$ is *false* on at least one line of the truth table then, for every line on the truth table where $\mathscr{S}$ is false, write down a disjunction $(\pm\mathscr{A}_1 \vee \ldots \vee \pm\mathscr{A}_n)$ which is *false* on (and only on) that line. Let $\mathscr{C}$ be the conjunction of all of these disjuncts; by construction, $\mathscr{C}$ is in CNF and $\mathscr{S}$ and $\mathscr{C}$ are logically equivalent.

Practice exercises

A. Consider the following sentences:

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

For each sentence, find a logically equivalent sentence in DNF and one in CNF.

38.4 The expressive adequacy of TFL

Of our connectives, $\neg$ attaches to a single sentences, and the others all combine exactly two sentences. We may also introduce the idea of an n-place connective. For example, we could consider a three-place connective, '$\heartsuit$', and stipulate that it is to have the following characteristic truth table:

A	B	C	$\heartsuit(A, B, C)$
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	F

Probably this new connective would not correspond with any natural English expression (at least not in the way that '$\wedge$' corresponds with 'and'). But a

question arises: if we wanted to employ a connective with this characteristic truth table, must we add a *new* connective to TFL? Or can we get by with the connectives we *already have*?

Let us make this question more precise. Say that some connectives are JOINTLY EXPRESSIVELY ADEQUATE *iff*, for any possible truth table, there is a sentence containing only those connectives with that truth table.

The general point is, when we are armed with some jointly expressively adequate connectives, no characteristic truth table lies beyond our grasp. And in fact, we are in luck.

Expressive Adequacy Theorem. The connectives of TFL are jointly expressively adequate. Indeed, the following pairs of connectives are jointly expressively adequate:

1. '$\neg$' and '$\vee$'
2. '$\neg$' and '$\wedge$'
3. '$\neg$' and '$\rightarrow$'

Given any truth table, we can use the method of proving the DNF Theorem (or the CNF Theorem) via truth tables, to write down a scheme which has the same characteristic truth table. For example, employing the truth table method for proving the DNF Theorem, we find that the following scheme has the same characteristic truth table as $\heartsuit(A, B, C)$, above:

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

It follows that the connectives of TFL are jointly expressively adequate. We now prove each of the subsidiary results.

Subsidiary Result 1: expressive adequacy of '$\neg$' and '$\vee$'. Observe that the scheme that we generate, using the truth table method of proving the DNF Theorem, will only contain the connectives '$\neg$', '$\wedge$' and '$\vee$'. So it suffices to show that there is an equivalent scheme which contains only '$\neg$' and '$\vee$'. To show do this, we simply consider that

$$(\mathscr{A} \wedge \mathscr{B}) \quad \text{and} \quad \neg(\neg\mathscr{A} \vee \neg\mathscr{B})$$

are logically equivalent.

Subsidiary Result 2: expressive adequacy of '$\neg$' and '$\wedge$'. Exactly as in Subsidiary Result 1, making use of the fact that

$$(\mathscr{A} \vee \mathscr{B}) \quad \text{and} \quad \neg(\neg\mathscr{A} \wedge \neg\mathscr{B})$$

are logically equivalent.

Subsidiary Result 3: expressive adequacy of '$\neg$' and '$\rightarrow$'. Exactly as in Subsidiary Result 1, making use of these equivalences instead:

$$(\mathscr{A} \vee \mathscr{B}) \quad \text{and} \quad (\neg\mathscr{A} \rightarrow \mathscr{B})$$
$$(\mathscr{A} \wedge \mathscr{B}) \quad \text{and} \quad \neg(\mathscr{A} \rightarrow \neg\mathscr{B})$$

Alternatively, we could simply rely upon one of the other two subsidiary results, and (repeatedly) invoke only one of these two equivalences.

In short, there is never any *need* to add new connectives to TFL. Indeed, there is already some redundancy among the connectives we have: we could have made do with just two connectives, if we had been feeling really austere.

38.5 Individually expressively adequate connectives

In fact, some two-place connectives are *individually* expressively adequate. These connectives are not standardly included in TFL, since they are rather cumbersome to use. But their existence shows that, if we had wanted to, we could have defined a truth-functional language that was expressively adequate, which contained only a single primitive connective.

The first such connective we will consider is '↑', which has the following characteristic truth table.

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

This is often called 'the Sheffer stroke', after Henry Sheffer, who used it to show how to reduce the number of logical connectives in Russell and Whitehead's *Principia Mathematica*.[1] (In fact, Charles Sanders Peirce had anticipated Sheffer by about 30 years, but never published his results.)[2] It is quite common, as well, to call it 'nand', since its characteristic truth table is the negation of the truth table for '∧'.

'↑' is expressively adequate all by itself.

The Expressive Adequacy Theorem tells us that '¬' and '∨' are jointly expressively adequate. So it suffices to show that, given any scheme which contains only those two connectives, we can rewrite it as a logically equivalent scheme which contains only '↑'. As in the proof of the subsidiary cases of the Expressive Adequacy Theorem, then, we simply apply the following equivalences:

$$\neg\mathscr{A} \quad \text{and} \quad (\mathscr{A} \uparrow \mathscr{A})$$
$$(\mathscr{A} \vee \mathscr{B}) \quad \text{and} \quad ((\mathscr{A} \uparrow \mathscr{A}) \uparrow (\mathscr{B} \uparrow \mathscr{B}))$$

to the Subsidiary Result 1.

Similarly, we can consider the connective '↓':

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

This is sometimes called the 'Peirce arrow' (Peirce himself called it 'ampheck'). More often, though, it is called 'nor', since its characteristic truth table is the negation of '∨', that is, of 'neither ... nor ...'.

[1] Sheffer, 'A Set of Five Independent Postulates for Boolean Algebras, with application to logical constants,' (1913, *Transactions of the American Mathematical Society* 14.4)

[2] See Peirce, 'A Boolian Algebra with One Constant', which dates to c.1880; and Peirce's *Collected Papers*, 4.264–5.

'$\downarrow$' is expressively adequate all by itself.

As in the previous result for $\uparrow$, although invoking the equivalences:

$$\begin{array}{rcl} \neg\mathscr{A} & \text{and} & (\mathscr{A} \downarrow \mathscr{A}) \\ (\mathscr{A} \wedge \mathscr{B}) & \text{and} & ((\mathscr{A} \downarrow \mathscr{A}) \downarrow (\mathscr{B} \downarrow \mathscr{B})) \end{array}$$

and Subsidiary Result 2.

38.6 Failures of expressive adequacy

In fact, the *only* two-place connectives which are individually expressively adequate are '$\uparrow$' and '$\downarrow$'. But how would we show this? More generally, how can we show that some connectives are *not* jointly expressively adequate?

The obvious thing to do is to try to find some truth table which we *cannot* express, using just the given connectives. But there is a bit of an art to this.

To make this concrete, let's consider the question of whether '$\vee$' is expressively adequate all by itself. After a little reflection, it should be clear that it is not. In particular, it should be clear that any scheme which only contains disjunctions cannot have the same truth table as negation, i.e.:

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

The intuitive reason, why this should be so, is simple: the top line of the desired truth table needs to have the value False; but the top line of any truth table for a scheme which *only* contains $\vee$ will always be True. The same is true for $\wedge$, $\rightarrow$, and $\leftrightarrow$.

'$\vee$', '$\wedge$', '$\rightarrow$', and '$\leftrightarrow$' are not expressively adequate by themselves.

In fact, the following is true:

The *only* two-place connectives that are expressively adequate by themselves are '$\uparrow$' and '$\downarrow$'.

This is of course harder to prove than for the primitive connectives. For instance, the "exclusive or" connective does not have a T in the first line of its characteristic truth table, and so the method used above no longer suffices to show that it cannot express all truth tables. It is also harder to show that, e.g., '$\leftrightarrow$' and '$\neg$' *together* are not expressively adequate.

Appendices

Symbolic notation A

1.1 Alternative nomenclature

Truth-functional logic. TFL goes by other names. Sometimes it is called *sentential logic*, because it deals fundamentally with sentences. Sometimes it is called *propositional logic*, on the idea that it deals fundamentally with propositions. We have stuck with *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.

First-order logic. FOL goes by other names. Sometimes it is called *predicate logic*, because it allows us to apply predicates to objects. Sometimes it is called *quantified logic*, because it makes use of quantifiers.

Formulas. Some texts call formulas *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 'formula'.

In §6, we defined *sentences* of TFL. These are also sometimes called 'formulas' (or 'well-formed formulas') since in TFL, unlike FOL, there is no distinction between a formula and a 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 FOL, 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'. (Its etymology still lingers in certain fonts, particularly in italic fonts; thus an italic ampersand might appear as '*&*'.) 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 two common symbols for the material conditional: the *arrow*, '$\rightarrow$', and the *hook*, '$\supset$'.

Material Biconditional. The *double-headed arrow*, '$\leftrightarrow$', is used in systems that use the arrow to represent the material conditional. Systems that use the hook for the conditional typically use the *triple bar*, '$\equiv$', for the biconditional.

Quantifiers. The universal quantifier is typically symbolized as 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	$\neg$, $\sim$
conjunction	$\wedge$, &, $\bullet$
disjunction	$\vee$
conditional	$\rightarrow$, $\supset$
biconditional	$\leftrightarrow$, $\equiv$
universal quantifier	$\forall x$, (x)

Alternative proof systems B

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 disjunction, negation, and the quantifiers. We will also explain why we have made the choices that we have.

2.1 Alternative disjunction elimination

Some systems take DS as their basic rule for disjunction elimination. Such systems can then treat the $\vee$E rule as a derived rule. For they might offer the following proof scheme:

m	$\mathscr{A} \vee \mathscr{B}$	
o	$\mathscr{A} \vee \neg\mathscr{A}$	LEM
i	$\quad \mathscr{A}$	
j	$\quad \mathscr{C}$	
k	$\quad \mathscr{B}$	
l	$\quad \mathscr{C}$	
n	$\mathscr{A} \to \mathscr{C}$	$\to$I i–j
$n+1$	$\mathscr{B} \to \mathscr{C}$	$\to$I k–l
$n+2$	$\quad \mathscr{C}$	
$n+3$	$\quad \mathscr{C}$	R $n+2$
$n+4$	$\quad \neg\mathscr{C}$	
$n+5$	$\quad\quad \mathscr{A}$	
$n+6$	$\quad\quad \mathscr{C}$	$\to$E n, $n+5$
$n+7$	$\quad\quad \bot$	$\bot$I $n+6$, $n+4$
$n+8$	$\quad \neg\mathscr{A}$	$\neg$I $n+5$–$n+7$
$n+9$	$\quad \mathscr{B}$	DS m, $n+8$
$n+10$	$\quad \mathscr{C}$	$\to$E $n+1$, $n+9$
$n+11$	$\mathscr{C}$	$\vee$E o, $n+2$–$n+3$, $n+4$–$n+10$

So why did we choose to take $\vee$E as basic, rather than DS?[1] Our reasoning is that DS involves the use of '$\neg$' in the statement of the rule. It is in some sense 'cleaner' for our disjunction elimination rule to avoid mentioning *other* connectives.

2.2 Alternative negation rules

Some systems take the following rule as their basic negation introduction rule:

m	$\quad \mathscr{A}$	
$n-1$	$\quad \mathscr{B}$	
n	$\quad \neg\mathscr{B}$	
	$\neg\mathscr{A}$	$\neg$I* m–n

and the following as their basic negation elimination rule:

[1] P.D. Magnus's original version of this book went the other way.

m	$\quad\mid\ \neg\mathscr{A}$	
$n-1$	$\quad\mid\ \mathscr{B}$	
n	$\quad\mid\ \neg\mathscr{B}$	
	$\mathscr{A}$	$\neg$E* m–n

Using these two rules, we could have derived all of the rules governing negation and contradiction that we have taken as basic (i.e. $\bot$I, $\bot$E, $\neg$I and LEM). Indeed, we could have avoided all use of the symbol '$\bot$' altogether. Negation would have had a single introduction and elimination rule, and would have behaved much more like the other connectives.

The resulting system would have had fewer rules than ours. So why did we chose to separate out contradiction, and to use an explicit rule LEM?[2]

Our first reason is that adding the symbol '$\bot$' to our natural deduction system makes proofs considerably easier to work with.

Our second reason is that a lot of fascinating philosophical discussion has focussed on the acceptability or otherwise of *law of excluded middle* (i.e. LEM) and *explosion* (i.e. $\bot$E). By treating these as separate rules in the proof system, we will be in a better position to engage with that philosophical discussion. In particular: having invoked these rules explicitly, it will be much easier for us to know what a system which lacked these rules would look like.

2.3 Alternative quantification rules

An alternative approach to the quantifiers is to take as basic the rules for $\forall$I and $\forall$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]

Taking only these rules as basic, we could have derived the $\exists$I and $\exists$E rules provided in §30. To derive the $\exists$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:

m	$\mathscr{A}(\ldots c \ldots c \ldots)$
k	$\exists x\mathscr{A}(\ldots x \ldots c \ldots)$

This is not yet permitted, since in this new system, we do not have the $\exists$I rule. We can, however, offer the following:

[2] Again, 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.

m	$\mathscr{A}(\ldots c \ldots c \ldots)$	
$m+1$	$\quad \neg\exists x\mathscr{A}(\ldots x \ldots c \ldots)$	
$m+2$	$\quad \forall x\neg\mathscr{A}(\ldots x \ldots c \ldots)$	CQ $m+1$
$m+3$	$\quad \neg\mathscr{A}(\ldots c \ldots c \ldots)$	$\forall$E $m+2$
$m+4$	$\quad \bot$	$\bot$I m, $m+3$
$m+5$	$\neg\neg\exists x\mathscr{A}(\ldots x \ldots c \ldots)$	$\neg$I $m+1$–$m+4$
$m+6$	$\exists x\mathscr{A}(\ldots x \ldots c \ldots)$	DNE $m+5$

To derive the $\exists$E rule is rather more subtle. This is because the $\exists$E rule has an important constraint (as, indeed, does the $\forall$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:

m	$\exists x\mathscr{A}(\ldots x \ldots x \ldots)$
i	$\quad \mathscr{A}(\ldots c \ldots c \ldots)$
j	$\quad \mathscr{B}$
k	$\mathscr{B}$

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 $\exists$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:

m	$\exists x\mathscr{A}(\ldots x \ldots x \ldots)$	
i	$\quad \mathscr{A}(\ldots c \ldots c \ldots)$	
j	$\quad \mathscr{B}$	
k	$\mathscr{A}(\ldots c \ldots c \ldots) \to \mathscr{B}$	$\to$I i–j
$k+1$	$\quad \neg\mathscr{B}$	
$k+2$	$\quad \neg\mathscr{A}(\ldots c \ldots c \ldots)$	MT k, $k+1$
$k+3$	$\quad \forall x\neg\mathscr{A}(\ldots x \ldots x \ldots)$	$\forall$I $k+2$
$k+4$	$\quad \neg\exists x\mathscr{A}(\ldots x \ldots x \ldots)$	CQ $k+3$
$k+5$	$\quad \bot$	$\bot$I m, $k+4$
$k+6$	$\neg\neg\mathscr{B}$	$\neg$I $k+1$–$k+5$
$k+7$	$\mathscr{B}$	DNE $k+6$

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 §32.

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 ∃I and ∃E that we have offered in this section, we invoked DNE. This is a derived rule, whose derivation essentially depends upon the use of LEM. But, as we mentioned earlier, LEM is a contentious rule. So, if we want to move to a system which abandons LEM, 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 LEM, we will be *unable* to derive the CQ rule which moves from $\neg\forall x\mathscr{A}$ to $\exists x\neg\mathscr{A}$.)

Quick reference C

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	$\neg P$
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 $\neg\exists x(Fx \wedge Gx)$

Identity

Only c is G	$\forall x(Gx \leftrightarrow x = c)$
Everything besides c is G	$\forall x(\neg x = c \rightarrow Gx)$
The F is G	$\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge Gx)$
It is not the case that the F is G	$\neg\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge Gx)$
The F is non-G	$\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge \neg Gx)$

3.3 Using identity to symbolize quantities

There are at least _____ Fs.

one: $\exists x Fx$
two: $\exists x_1 \exists x_2 (Fx_1 \wedge Fx_2 \wedge \neg x_1 = x_2)$
three: $\exists x_1 \exists x_2 \exists x_3 (Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge \neg x_1 = x_2 \wedge \neg x_1 = x_3 \wedge \neg x_2 = x_3)$
four: $\exists x_1 \exists x_2 \exists x_3 \exists x_4 (Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge Fx_4 \wedge$
$\neg x_1 = x_2 \wedge \neg x_1 = x_3 \wedge \neg x_1 = x_4 \wedge \neg x_2 = x_3 \wedge \neg x_2 = x_4 \wedge \neg x_3 = x_4)$
n: $\exists x_1 \ldots \exists x_n (Fx_1 \wedge \ldots \wedge Fx_n \wedge \neg x_1 = x_2 \wedge \ldots \wedge \neg x_{n-1} = x_n)$

There are at most _____ Fs.

One way to say 'there are at most n Fs' is to put a negation sign in front of the symbolization for 'there are at least $n+1$ Fs'. Equivalently, we can offer:

one: $\forall x_1 \forall x_2 \big[(Fx_1 \wedge Fx_2) \rightarrow x_1 = x_2\big]$
two: $\forall x_1 \forall x_2 \forall x_3 \big[(Fx_1 \wedge Fx_2 \wedge Fx_3) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3)\big]$
three: $\forall x_1 \forall x_2 \forall x_3 \forall x_4 \big[(Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge Fx_4) \rightarrow$
$(x_1 = x_2 \vee x_1 = x_3 \vee x_1 = x_4 \vee x_2 = x_3 \vee x_2 = x_4 \vee x_3 = x_4)\big]$
n: $\forall x_1 \ldots \forall x_{n+1} \big[(Fx_1 \wedge \ldots \wedge Fx_{n+1}) \rightarrow (x_1 = x_2 \vee \ldots \vee x_n = x_{n+1})\big]$

There are exactly _____ Fs.

One way to say 'there are exactly n Fs' is to conjoin two of the symbolizations above and say 'there are at least n Fs and there are at most n Fs.' The following equivalent formulae are shorter:

zero: $\forall x \neg Fx$
one: $\exists x \big[Fx \wedge \forall y (Fy \rightarrow x = y)\big]$
two: $\exists x_1 \exists x_2 \big[Fx_1 \wedge Fx_2 \wedge \neg x_1 = x_2 \wedge \forall y \big(Fy \rightarrow (y = x_1 \vee y = x_2)\big)\big]$
three: $\exists x_1 \exists x_2 \exists x_3 \big[Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge \neg x_1 = x_2 \wedge \neg x_1 = x_3 \wedge \neg x_2 = x_3 \wedge$
$\forall y \big(Fy \rightarrow (y = x_1 \vee y = x_2 \vee y = x_3)\big)\big]$
n: $\exists x_1 \ldots \exists x_n \big[Fx_1 \wedge \ldots \wedge Fx_n \wedge \neg x_1 = x_2 \wedge \ldots \wedge \neg x_{n-1} = x_n \wedge$
$\forall y \big(Fy \rightarrow (y = x_1 \vee \ldots \vee y = x_n)\big)\big]$

3.4 Basic deduction rules for TFL

Reiteration

m	$\mathscr{A}$	
	$\mathscr{A}$	R m

Conjunction

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

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

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

Conditional

m	$\quad \mathscr{A}$	
	$\quad \vdots$	
n	$\quad \mathscr{B}$	
	$\mathscr{A} \to \mathscr{B}$	$\to$I m–n

m	$\mathscr{A} \to \mathscr{B}$	
n	$\mathscr{A}$	
	$\mathscr{B}$	$\to$E m, n

Biconditional

m	$\mathscr{A} \leftrightarrow \mathscr{B}$	
	$(\mathscr{A} \to \mathscr{B}) \wedge (\mathscr{B} \to \mathscr{A})$	$\leftrightarrow$E m

m	$(\mathscr{A} \to \mathscr{B}) \wedge (\mathscr{B} \to \mathscr{A})$	
	$\mathscr{A} \leftrightarrow \mathscr{B}$	$\leftrightarrow$I m

Contradiction

m	$\mathscr{A}$	
n	$\neg\mathscr{A}$	
	$\bot$	$\bot$I m, n

m	$\bot$	
	$\mathscr{A}$	$\bot$E m

Negation

m	$\quad \mathscr{A}$	
	$\quad \vdots$	
n	$\quad \bot$	
	$\neg\mathscr{A}$	$\neg$I m–n

Disjunction

m	$\mathscr{A}$	
	$\mathscr{A} \vee \mathscr{B}$	$\vee$I m

m	$\mathscr{A}$	
	$\mathscr{B} \vee \mathscr{A}$	$\vee$I m

m	$\mathscr{A} \vee \mathscr{B}$	
i	$\quad \mathscr{A}$	
	$\quad \vdots$	
j	$\quad \mathscr{C}$	
k	$\quad \mathscr{B}$	
	$\quad \vdots$	
l	$\quad \mathscr{C}$	
	$\mathscr{C}$	$\vee$E m, i–j, k–l

Law of Excluded Middle

	$\mathscr{A} \vee \neg\mathscr{A}$	LEM

3.5 Derived rules for TFL

Disjunctive syllogism

m	$\mathscr{A} \lor \mathscr{B}$	
n	$\neg\mathscr{A}$	
	$\mathscr{B}$	DS m, n

m	$\mathscr{A} \lor \mathscr{B}$	
n	$\neg\mathscr{B}$	
	$\mathscr{A}$	DS m, n

Modus Tollens

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

Double-negation elimination

m	$\neg\neg\mathscr{A}$	
	$\mathscr{A}$	DNE m

Proof by Contradiction

m	$\quad \neg\mathscr{A}$	
	$\quad \vdots$	
n	$\quad \bot$	
	$\mathscr{A}$	IP m–n

De Morgan Rules

m	$\neg(\mathscr{A} \lor \mathscr{B})$	
	$\neg\mathscr{A} \land \neg\mathscr{B}$	DeM m

m	$\neg\mathscr{A} \land \neg\mathscr{B}$	
	$\neg(\mathscr{A} \lor \mathscr{B})$	DeM m

m	$\neg(\mathscr{A} \land \mathscr{B})$	
	$\neg\mathscr{A} \lor \neg\mathscr{B}$	DeM m

m	$\neg\mathscr{A} \lor \neg\mathscr{B}$	
	$\neg(\mathscr{A} \land \mathscr{B})$	DeM m

3.6 Basic deduction rules for FOL

Universal elimination

m	$\forall x \mathscr{A}(\ldots x \ldots x \ldots)$	
	$\mathscr{A}(\ldots c \ldots c \ldots)$	$\forall$E m

Universal introduction

m	$\mathscr{A}(\ldots c \ldots c \ldots)$	
	$\forall x \mathscr{A}(\ldots x \ldots x \ldots)$	$\forall$I m

c must not occur in any undischarged assumption
x must not occur in $\mathscr{A}(\ldots c \ldots c \ldots)$

Identity introduction

	$c = c$	=I

Identity elimination

m	$a = b$	
n	$\mathscr{A}(\ldots a \ldots a \ldots)$	
	$\mathscr{A}(\ldots b \ldots a \ldots)$	=E m, n

m	$a = b$	
n	$\mathscr{A}(\ldots b \ldots b \ldots)$	
	$\mathscr{A}(\ldots a \ldots b \ldots)$	=E m, n

Existential introduction

m	$\mathscr{A}(\ldots c \ldots c \ldots)$	
	$\exists x \mathscr{A}(\ldots x \ldots c \ldots)$	$\exists$I m

x must not occur in $\mathscr{A}(\ldots c \ldots c \ldots)$

Existential elimination

m	$\exists x \mathscr{A}(\ldots x \ldots x \ldots)$	
i	$\quad \mathscr{A}(\ldots c \ldots c \ldots)$	
j	$\quad \mathscr{B}$	
	$\mathscr{B}$	$\exists$E m, i–j

c must not occur in any undischarged assumption, in $\exists x \mathscr{A}(\ldots x \ldots x \ldots)$, or in $\mathscr{B}$

3.7 Derived rules for FOL

m	$\forall x \neg \mathscr{A}$	
	$\neg \exists x \mathscr{A}$	CQ m

m	$\exists x \neg \mathscr{A}$	
	$\neg \forall x \mathscr{A}$	CQ m

m	$\neg \exists x \mathscr{A}$	
	$\forall x \neg \mathscr{A}$	CQ m

m	$\neg \forall x \mathscr{A}$	
	$\exists x \neg \mathscr{A}$	CQ m

Glossary

antecedent The sentence on the left side of a conditional.

argument a connected series of sentences, divided into premises and conclusion.

atomic sentence A sentence used to represent a basic sentence; a single letter in TFL, or a predicate symbol followed by names in FOL.

biconditional The symbol $\leftrightarrow$, used to represent words and phrases that function like the English phrase "if and only if"; or a sentence formed using this connective..

() .

complete truth table A table that gives all the possible truth values for a sentence of TFL or sentences in TFL, with a line for every possible valuation of all atomic sentences.

completeness A property held by logical systems if and only if $\vDash$ implies $\vdash$.

conclusion the last sentence in an argument.

conclusion indicator a word or phrase such as "therefore" used to indicate that what follows is the conclusion of an argument.

conditional The symbol $\rightarrow$, used to represent words and phrases that function like the English phrase "if ... then ..."; a sentence formed by using this symbol.

conjunct A sentence joined to another by a conjunction.

conjunction The symbol $\wedge$, used to represent words and phrases that function like the English word "and"; or a sentence formed using that symbol.

conjunctive normal form (DNF) a sentence which is a conjunction of disjunctions of atomic sentences or negated atomic sentences.

connective A logical operator in TFL used to combine atomic sentences into larger sentences.

consequent The sentence on the right side of a conditional.

contingent sentence A sentence that is neither a necessary truth nor a necessary falsehood; a sentence that in some situations is true and in others false.

contradiction (of FOL) A that is false in every interpretation.

contradiction (of TFL) A sentence that has only Fs in the column under the main logical operator of its complete truth table; a sentence that is false on every valuation.

disjunct A sentence joined to another by a disjunction.

disjunction The connective $\vee$, used to represent words and phrases that function like the English word "or" in its inclusive sense; or a sentence formed by using this connective.

disjunctive normal form (DNF) a sentence which is a disjunction of conjunctions of atomic sentences or negated atomic sentences.

domain the collection of objects assumed for a symbolization in FOL, or that gives the range of the quantifiers in an interpretation.

empty predicate a predicate that applies to no object in the domain.

entailment (in TFL) Some sentences entail another if and only if no valuation makes all former sentences true and the latter false.

existential quantifier the symbol $\exists$ of FOL used to symbolize existence; $\exists x\, Fx$ is true iff at least one member of the domain is F.

expressive adequacy property of a collection of connectives which holds iff every possible truth table is the truth table of a sentence involving only those connectives.

formation tree A tree showing the structure of a sentence and its subsentences..

() .

() .

interpretation a specification of a domain together with the objects the names pick out and which objects the predicates are true of.

invalid A property of arguments that holds when it is possible for the premises to be true without the conclusion being true; the opposite of valid.

joint possibility A property possessed by some sentences when they can all be true at the same time.

logical consistency (in FOL) A property held by if and only if some interpretation makes all the sentences true.

logical consistency (in TFL) A property held by sentences if and only if the complete truth table for those sentences contains one line on which all the sentences are true, i.e., if some valuation makes all the sentences true.

logical equivalence (in FOL) A property held by pairs of if and only if the sentences have the same truth value in every interpretation..

logical equivalence (in TFL) A property held by pairs of sentences if and only if the complete truth table for those sentences has identical columns under the two main logical operators, i.e., if the sentences have the same truth value on every valuation.

logical truth A that is true in every interpretation.

logical validity (in FOL) A property held by arguments if and only if no interpretation makes all premises true and the conclusion false.

logical validity (in TFL) A property held by arguments if and only if the complete truth table for the argument contains no rows where the premises are all true and the conclusion false, i.e., if no valuation makes all premises true and the conclusion false.

main connective The last connective that you add when you assemble a sentence using the recursive definition..

metalanguage The language logicians use to talk about the object language. In this textbook, the metalanguage is English, supplemented by certain symbols like metavariables and technical terms like "valid.".

metavariables A variable in the metalanguage that can represent any sentence in the object language..

name a symbol of FOL used to pick out an object of the domain.

necessary equivalence A property held by a pair of sentences that must always have the same truth value.

necessary falsehood A sentence that must be false.

necessary truth A sentence that must be true.

negation The symbol $\neg$, used to represent words and phrases that function like the English word "not".

object language A language that is constructed and studied by logicians. In this textbook, the object languages are TFL and FOL..

predicate a symbol of FOL used to symbolize a property or relation.

premise a sentence in an argument other than the conclusion.

premise indicator a word or phrase such as "because" used to indicate that what follows is the premise of an argument.

provable equivalence A property held by pairs of statements if and only if there is a derivation which takes you from each one to the other one.

provable inconsistency Sentences are provably inconsistent iff a contradiction can be derived from them.

scope The sentences that are joined by a connective. These are the sentences the connective was applied to when the sentence was assembled using a recursive definition..

() .

sentence of TFL A string of symbols in TFL that can be built up according to the recursive rules given on p. 30..

sound A property of arguments that holds if the argument is valid and has all true premises.

soundness A property held by logical systems if and only if $\vdash$ implies $\vDash$.

substitution instance the result of replacing every occurrence of a in a with a name.

symbolization key A list that shows which English sentences are represented by which atomic sentences in TFL.

tautology A sentence that has only Ts in the column under the main logical operator of its complete truth table; a sentence that is true on every valuation.

() .

theorem A sentence that can be proved without any premises.

truth value One of the two logical values sentences can have: True and False.

truth-functional connective an operator that builds larger sentences out of smaller ones and fixes the truth value of the resulting sentence based only on the truth value of the component sentences.

universal quantifier the symbol $\forall$ of FOL used to symbolize generality; $\forall x\, Fx$ is true iff every member of the domain is F.

valid A property of arguments where it is impossible for the premises to be true and the conclusion false.

valuation An assignment of truth values to particular atomic sentence of TFLs.

variable a symbol of FOL used following quantifiers and as placeholders in atomic formulas; lowercase letters between s and z.

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.