forallχ

University of York
Reason & Argument 2023-4

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

Modified for Cambridge by:
Tim Button
University College London

Further modified for York by:
Robert Trueman
University of York

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

Tim Button would like to thank P.D. Magnus for his extraordinary act of generosity, in making forallχ available to everyone. Thanks also to Alfredo Manfredini Böhm, Sam Brain, Felicity Davies, Emily Dyson, Phoebe Hill, Richard Jennings, Justin Niven, and Igor Stojanovic for noticing errata in earlier versions.

Robert Trueman would also like to thank P.D. Magnus, and Tim Button as well.

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

Contents

Chapter 1

Key notions

Arguments 1

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

In everyday life, we sometimes use the word 'argument' to talk about angry shouting matches. But logic is not concerned with teeth-gnashing and hair-pulling. They are not arguments, in our sense; they are unhappy disagreements.

The kind of arguments that we are interested in are more like this:

You are taking *Reason and Argument*.
Everyone who takes *Reason and Argument* loves logic.
So: You love logic.

We have here a series of sentences. The word 'So' on the third line indicates that the final sentence expresses the *conclusion* of the argument. The two sentences before that express *premises* of the argument. If you believe the premises, then the argument provides you with a reason to believe the conclusion.

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

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:

Hadi was laughing; so Tracy must have told him a good joke.

This argument has one premise followed by a conclusion.

Many arguments start with premises, and end with a conclusion. But not all of them. Sometimes we put the conclusion at the beginning:

Simon must be annoyed at Daniel. After all, Daniel kept Simon up all night with his snoring, and Simon always gets annoyed when Daniel does that.

And sometimes we even put the conclusion in the middle:

Erica cooked dinner. So there won't be any meat in the meal, because Erica is vegetarian.

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

And 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.

Practice exercises

At the end of some sections, there are problems that review and explore the material covered in the chapter. You should work through all of these problems, because getting good at logic is more about learning a new way of thinking, than it is about just memorising 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. And 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:

The Moon is made of Swiss cheese.
So: 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, it has two problems. First: the argument's (only) premise is obviously false. The Moon isn't made of cheese. (It's basically a big rock. Or maybe an alien observatory—see YouTube.) Second: the conclusion does not follow from the premise of the argument. Even if the Moon were made of Swiss cheese, we would not be able to draw any conclusion about Dali's predilection for poker.

In general, then, an argument can go wrong in two ways:

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

We usually care a lot about whether the premises of an argument are true. But that is normally a task best left to experts in the relevant field: we might need a scientist, or a historian, or a literary theorist, or whoever else to tell us whether the premises of an argument are true. 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, as logicians, we are interested in whether or not a conclusion *follows from* some premises. But don't let yourself slip into saying 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. But what, exactly, does that mean? Consider the following argument:

You are reading a logic textbook.
So: You are a logic student.

That looks like a pretty good argument. The premise definitely gives us some reason to believe the conclusion. For many purposes, this might be enough for us to say that the conclusion 'follows' from the premises. But in formal logic, we set the bar higher than that. Notice that it is possible for the premise to be true and the conclusion to be false: glancing at this textbook over your friend's shoulder wouldn't make you a logic student! In formal logic, then, we refuse to say that a conclusion follows from some premises unless it is impossible for the premises to be true and the conclusion be false. We capture this in the following definition:

> An argument is VALID if and only if it is impossible for all of the premises to be true and the conclusion to be false.
>
> An argument is INVALID if and only if it is not valid.

Validity is the *gold standard* for arguments. If an argument is valid, then the truth of its premises would be enough to *guarantee* the truth of its conclusion. We've already seen an example of a valid argument:

You are taking *Reason and Argument*.
Everyone who takes *Reason and Argument* loves logic.
So: You love logic.

If the premises of this argument are true, then the conclusion just has to be true too. Now, I have to admit, I don't know whether the premises are true. (I'm pretty confident about the second premise, but I can't be sure of the first: you might be one of those people glancing over a logic student's shoulder!) But that doesn't matter: *if* the premises are true, then the conclusion *must* be true, and so the argument is valid. We can even have valid arguments with obviously false premises:

Oranges are stringed instruments.
Stringed instruments are mammals.
So: Oranges are mammals.

The premises of this argument are ridiculous, and so is the conclusion. Nevertheless, the ridiculous conclusion does follow from the ridiculous premises. *If* both premises were true, *then* the conclusion would have to be true too. So the argument is valid.

So, 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.
So: 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. But, of course, we would ideally like our arguments to have true premises. So we will say that an argument is SOUND if and only if it is both valid and all of its premises are true.

There's a pretty good sense in which sound arguments are better than merely valid ones: if an argument is valid, you have a guarantee that the conclusion is true, *if the premises are*; but if an argument is sound, you have a guarantee that the conclusion is true, *full stop*. However, as we discussed earlier, it isn't usually the job of a logician to figure out if the premises of an argument are true or false. So, for the most part, we logicians have to settle for figuring out whether arguments are valid, and leave the question of their soundness for someone else.

2.3 Inductive arguments

We noted earlier that there are arguments which seem to be 'good' in some sense, but which are not valid. Consider this one:

In January 2015, it rained in London.
In January 2016, it rained in London.
In January 2017, it rained in London.
In January 2018, it rained in London.
So: 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 2019, it rained in London; In January 2020. . . . But, however many premises of this form we add, the argument will remain invalid. Even if it has rained in London in every January thus far, it remains *possible* that London will stay dry next January.

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

Practice exercises

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

1. Hypatia is a mathematician.
2. All mathematicians are carrots.

So: Therefore, Hypatia is a carrot.

1. Abe Lincoln was either 5ft tall or he was once president.
2. Abe Lincoln was never president.

So: Abe Lincoln was 5ft tall.

1. If Ingrid trained hard, then she will win the race.
2. Ingrid did not train hard.

So: Ingrid will not win the race.

1. Hugh Jackman was born in either France or Luxemborg.
2. Hugh Jackman was not born in Luxemborg.

So: Hugh Jackman was born in 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.

So: The world will not end today.

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. A sound argument with a false conclusion?
5. An invalid argument that can be made valid by the addition of a new premise?
6. 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 Truth-values

As we said in §1, arguments consist of premises and a conclusion. Note that many kinds of English sentence cannot be used to express premises or conclusions of arguments. For example:

- **Questions**, e.g. 'Are you feeling sleepy?'
- **Imperatives**, e.g. 'Wake up!'
- **Exclamations**, e.g. 'Ouch!'

The common feature of these three kinds of sentence is that they are not *assertoric*: they cannot be true or false. It does not even make sense to ask whether a *question* is true (it only makes sense to ask whether the *answer* to a question is true).

The general point is that, the premises and conclusion of an argument must be capable of being *true* or *false*. In logic, we say that a sentence which is true or false has a TRUTH-VALUE: its truth-value is either True or False.

3.2 Consistency

Consider these two sentences:

B1. Either Jane is 6ft tall, or Jane is 5ft11in tall.
B2. Jane is not 6ft tall, and Jane is not 5ft11in tall.

Logic alone cannot tell us which, if either, of these sentences is true. Yet we can say that *if* the first sentence (B1) is true, *then* the second sentence (B2) must be false. And if B2 is true, then B1 must be false. It is impossible that both sentences are true together. These sentences are inconsistent with each other. And this motivates the following definition:

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

Using this new terminology, then, we can say that B1 and B2 are jointly inconsistent.

We can ask about the consistency 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 inconsistent. They cannot all be true together. (Note that the sentences G1, G3 and G4 are jointly inconsistent. But if sentences are already jointly inconsistent, adding an extra sentence to the mix will not make them consistent!)

3.3 Necessity and contingency

When we're 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.

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. We call sentences like 2 *necessary truths*:

> A sentence is a NECESSARY TRUTH if and only if it is impossible for it to be false.

You also 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. We call sentences like 3 *necessary falsehoods*.

> A sentence is a NECESSARY FALSEHOOD if and only if it is impossible for it to be true.

Something which is capable of being true or false, but which is neither necessarily true nor necessarily false, is CONTINGENT.

3.4 Possible worlds

Philosophers and logicians sometimes find it useful to speak about *possible worlds*. The idea is that, for each way the world *could* be, there is a possible world which *is* that way. So, for example, since you *could* have studied engineering rather than philosophy, there is a possible world where you *are* studying engineering.

We can use this way of speaking to offer alternative definitions of all the core concepts we have covered so far:

> A sentence is a NECESSARY TRUTH if and only if it is true in all possible worlds.
>
> A sentence is a NECESSARY FALSEHOOD if and only if it is false in all possible worlds.
>
> Some sentences are JOINTLY CONSISTENT if and only if there is some possible world where they are all true together.
>
> An argument is VALID if and only if there is no possible world where all of the premises are true and the conclusion is false.

'Possible world' talk can be very convenient, but be warned: philosophers disagree about the best way to understand it. Some philosophers think that we can get away with talking about possible worlds as a 'useful fiction' (a bit like the way that physicists get away with talking about frictionless planes). But other philosophers think that, if you want to talk about possible worlds, then you need to *really believe* in them. However, that is a topic for another module. (In particular, it's a topic for the second-year Metaphysics module...)

3.5 Explosion

Imagine an argument with jointly inconsistent premises, and any conclusion you like. Here's an example:

1. I am over 35 years old.
2. I am less than 35 years old.

So: You are loving this textbook.

Is this argument valid or invalid? Well, this might come as a surprise, but the answer is: *Valid!* An argument is valid if there is no possible world where the premises are all true and the conclusion is false. But since the premises are jointly inconsistent, there is no possible world where they are all true. So, trivially, there is no possible world where the premise are all true *and* the conclusion is false.

In general, then, any argument with jointly inconsistent premises is automatically valid, no matter what the conclusion is. Logicians call this the EXPLOSION rule, because inconsistent premises explode logical systems by entailing everything. (It used to have fancy old Latin names: sometimes *ex falso quodlibet*, and sometime *ex contradictione quodlibet*. But learning logic is hard enough without having to learn Latin too!)

Explosion can seem pretty dodgy when you're first introduced to it. However, it is built into our definition of validity. Now, some logicians and philosophers have developed new definitions of validity which don't lead to Explosion. (These new definitions are now called *paraconsistent*, in case you want to know what to google.) However, these alternative definitions of validity make everything more complicated, and always have odd features of their own. So, to keep things simple, we will stick with the classical definition of validity, and accept Explosion, at least for the time being.

And anyway, it turns out that Explosion isn't all bad. In fact, it can help us to explain why it's so important not to have inconsistent beliefs. As a rule of thumb, you should believe anything that follows from what you already believe. So if you had inconsistent beliefs, you would have to believe *everything*—even silly things, e.g. that the Moon is made of cheese—because everything follows from inconsistent premises. And *that's* why you shouldn't have inconsistent beliefs!

Practice exercises

A. For each of the following: Is it necessarily true, necessarily false, 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. Look back at the sentences G1–G4 in this section (about giraffes, gorillas and martians in the wild animal park), and consider each of the following:

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

Which are jointly consistent? Which are jointly inconsistent?

C. Could there be:

1. A valid argument, the conclusion of which is necessarily false?
2. An invalid argument, the conclusion of which is necessarily true?
3. Jointly consistent sentences, one of which is necessarily false?
4. Jointly inconsistent sentences, one of which is necessarily true?

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

Chapter 2

Truth-Functional Logic

First steps to symbolisation 4

4.1 Valid argument forms

Consider this argument:

It is raining outside.
If it is raining outside, then Simon is miserable.
So: Simon is miserable.

and another argument:

Sharon is an archaeologist.
If Sharon is an archaeologist, then Sharon tells Rob a lot about old pots.
So: Sharon tells Rob a lot about old pots.

Both arguments are valid, and there is a straightforward sense in which we can say that they share a common *form*. We might display their form like this:

A
If A, then B
So: B

This is an excellent form of argument. Absolutely any argument which has this form will automatically be valid. Because of that, we can call this argument form itself *valid*.

> An argument form is VALID if and only if all arguments of that form are valid.
>
> An argument form is INVALID if and only if it is not valid.

(We are here defining what it is for an *argument form* to be valid in terms of our earlier definition of what it is for an *argument* to be valid.) Another example might help to convey the idea. Consider these two valid arguments:

Iayla is either happy or sad.
Iayla is not sad.
So: Iayla is happy.

Dave is either in his office or in the pub.
Dave is not in the pub.
So: Dave is in his office.

These arguments have a different form from the one we just looked at. We can display this new form like this:

A or B
not-B
So: A

This is another valid argument form: *any* argument of this form will be valid. And there are lots of other valid argument forms too. One of the jobs of a logician is to identify which forms of argument are valid. This is a useful thing to do, because if you know that an argument has a valid form, then you know it must be a valid argument: it doesn't matter what the premises or the conclusion actually *say*; the argument is already valid *in virtue of its form*.

4.2 Non-formal validity

If an argument has a valid form then it is automatically valid. *But be careful!* It does not follow that if an argument *does not* have a valid form then it is *invalid*. Some arguments are valid, but not just in virtue of their form. Here's an example:

The ball is green all over.
So: The ball is not red all over.

This argument is valid: if the premise is true then the conclusion must be true too, since nothing can be red and green all over. But this argument does not have a valid *form*. We can show this by cooking up an invalid argument with the same form, which is pretty easy:

The ball is green all over.
So: The ball is not shiny all over.

This argument is invalid, since something can be green and shiny all over. But this invalid argument has just the same form as the preceding valid argument. What this tells us is that the preceding argument is not valid just because of its form; its validity also depends on the particular meanings of 'red' and 'green'.

So, it is important not to get carried away with formal validity: an argument can be valid even if it doesn't have a valid form. But, at the same time, it is also important not to let that put you off formal validity altogether. As we said before, it is still useful to figure out which forms of argument are valid, because arguments with valid forms are always valid arguments.

4.3 Atomic sentences

Our aim in this chapter will be to develop a formal language which allows us to symbolise many arguments in a way that shows whether they are valid in virtue of their form. That language will be *Truth-Functional Logic*, or TFL.

We started isolating the form of an argument, in §4.1, by replacing sentences with letters. Thus, in the first example of that section, we replaced 'It is raining outside' with 'A', and 'Simon is miserable' with 'B'.

Our formal language, TFL, pursues this idea absolutely ruthlessly. We start with some *atomic sentences*. These will be our basic building blocks, which we will then use to construct more complex sentences. We will use uppercase italic letters for atomic sentences of TFL:

$$A, B, C, \ldots, Z$$

Annoyingly, though, there are only twenty-six letters of the alphabet, and there is no limit to the number of atomic sentences that we might want to consider. So we will also allow ourselves to make new atomic sentences by attaching numerical subscripts to letters, like this:

$$A_1, B_{17}, Q_{58}, W_{254}, Z_{3064}, \ldots$$

We shall use atomic sentence to represent, or symbolise, certain English sentences. To do this, we provide a SYMBOLISATION KEY, such as the following:

A: It is raining outside
B: Simon is miserable

When we present this key, we are not fixing this symbolisation *once and for all*. We are just saying that, for the time being, we shall think of the atomic sentence of TFL, 'A', as symbolising the English sentence 'It is raining outside', and the atomic sentence of TFL, 'B', as symbolising the English sentence 'Simon is miserable'. Later, when we are dealing with different sentences or different arguments, we can provide a new symbolisation key; as it might be:

A: Sharon is an archaeologist
B: Sharon tells Rob a lot about old pots

But it is important to understand that whatever structure an English sentence might have is lost when it is symbolised 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.

5 Connectives

TFL starts with atomic sentences, but it also builds bigger, more complex sentences out of those atoms. It does this by combining sentences with *logical connectives*. There are five logical connectives in TFL. They are summarised in this table, and then explained in more detail through the rest of this section.

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

5.1 Negation

Consider how we might symbolise 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 symbolise sentence 1, we will need an atomic sentence. We might offer this symbolisation key:

B: Mary is in Barcelona.

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

> A sentence can be symbolised as $\neg\mathcal{A}$ 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 car can be fixed.
5. The car is unfixable.

6. The car is not unfixable.

Let us use the following representation key:

R: The car can be fixed.

Sentence 4 can now be symbolised by 'R'. Moving on to sentence 5: saying the car is unfixable means that it is not the case that the car can be fixed. So even though sentence 5 does not contain the word 'not', we shall symbolise it as follows: '$\neg R$'.

Sentence 6 can be paraphrased as 'It is not the case that the car is unfixable.' Which can again be paraphrased as 'It is not the case that it is not the case that the car can be fixed'. So we might symbolise this English sentence with the TFL sentence '$\neg\neg R$'. (In English, double-negations tend to cancel out: sentence 6 says something very similar to 'The car is fixable'. Later on, you will be able to show that there is a sense in which TFL also allows us to cancel double-negations. (In particular, '$\neg\neg R$' is *tautologically equivalent* to 'R', in the sense defined in §11.2.) But, the point remains, if we want to capture the logical form of 6, then '$\neg\neg R$' is the most revealing way of symbolising it.)

The above examples reveal that the prefix 'un-' often functions as a negation in English, as in: 'unfixable', 'uneaten', 'undiscovered', etc. However, we need to be careful. Sometimes the prefix 'un-' does more than merely negate. Consider these examples:

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

If we let the TFL sentence 'H' symbolise sentence 7, then we might think that sentence 8 should be symbolised as '$\neg H$'. But that would be a mistake. In ordinary English, we can say that Jane is neither happy nor unhappy, but in a state of blank indifference. What this shows is that sentence 8 is more than just the negation of sentence 7: there is more to being unhappy than just not being happy.

Unfortunately, this is not just a special problem with the prefix 'un-'. Natural languages like English can be very messy, and it is often impossible to give one simple rule to cover all the ways that a particular word is used. (That's why we need to invent artificial languages like TFL!) Ultimately, then, you will need to rely on your own good linguistic judgment: only symbolise a sentence as $\neg\mathcal{A}$ if you think that it says no more and no less than 'It is not the case that $\mathcal{A}$'.

5.2 Conjunction

Consider these sentences:

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

We will need separate atomic sentences of TFL to symbolise sentences 9 and 10:

A: Jia is athletic.
B: Barbara is athletic.

Sentence 9 can now be symbolised as 'A', and sentence 10 can be symbolised as 'B'. Sentence 11 roughly says 'A and B'. We need another symbol, to deal with 'and'. We will use '$\wedge$'. Thus we will symbolise 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 symbolise 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) symbolise such things in TFL.

Some more examples will bring out this point:

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

Sentence 12 is clearly a conjunction, but what are its conjuncts? The first conjunct is obviously 'Jia is athletic', so you might be tempted to try to symbolise 12 as something like 'A and energetic'. But that would be a mistake! In TFL, we can only ever use whole sentences as conjuncts. So what we are really need to do is add another sentence letter, 'C', and use it to symbolise 'Jia is energetic'. Now we can symbolise 12 as '$(A \wedge C)$'.

We need to break sentence 13 up in a similar way. First, we paraphrase it as 'Barbara is athletic, and Jia is athletic'. We then symbolise it in TFL as '$(B \wedge A)$', using the symbolisation key above.

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'. And now we can symbolise this with the TFL sentence '$(D \wedge \neg B)$', where 'D' is a new sentence letter symbolising 'Barbara is energetic'. Note that we have lost all sorts of nuance in this symbolisation. 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. The word 'but' suggests a contrast between the two conjuncts, but that is not something that TFL can handle. So, the first step in symbolising sentence 15 is just to ignore this subtlety, and to paraphrase it as '*Both* Barbara is athletic, *and* Jia is more athletic than Barbara'. (Notice that we once again replace the pronoun 'her' with 'Barbara'.) How should we deal with the second conjunct? We already have the sentence letter 'A', which is being used to symbolise 'Jia is athletic', and the sentence 'B' which is being used to symbolise 'Barbara is athletic'; but neither of these concerns their relative athleticity. So, to symbolise the entire sentence, we need a new sentence letter. Let the TFL sentence 'E' symbolise the English sentence 'Jia is more athletic than Barbara'. Now we can symbolise sentence 15 as '$(B \wedge E)$'.

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

You might be wondering why we have put brackets around our conjunctions. To see the important role that these brackets play, we just need to think about how negation might interact with conjunction. Take a look at these two sentences:

16. It's not the case that you will have both chips and mashed potato.
17. You will not have chips but you will have mashed potato.

Sentence 16 can be paraphrased as 'It is not the case that: both you will have chips and you will have mashed potato'. Using this symbolisation key:

P_1: You will have chips.
P_2: You will have mashed potato.

We would symbolise 'both you will have chips and you will have mashed potato' as '$(P_1 \wedge P_2)$'. To symbolise sentence 16, then, we simply negate the whole sentence, thus: '$\neg(P_1 \wedge P_2)$'.

Sentence 17 is a conjunction: you *will not* have chips, and you *will* have mashed potato. 'You will not have chips' is symbolised by '$\neg P_1$'. So to symbolise sentence 17 itself, we offer '$(\neg P_1 \wedge P_2)$'.

These English sentences are very different, and their symbolisations differ accordingly. In one of them, the entire conjunction is negated. In the other, just one conjunct is negated. We call this a difference in the *scopes* of the negations, and we use the brackets to display this difference in scope. Without brackets, we would just write '$\neg P_1 \wedge P_2$', and it wouldn't be clear if we were trying to symbolise 16 or 17.

5.3 Disjunction

Consider these sentences:

18. Either Denison will play golf with me, or he will watch movies.
19. Either Denison or Ellery will play golf with me.

For these sentences we can use this symbolisation key:

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

However, we will also need to introduce another new symbol, this time for 'or'. Sentence 18 is symbolised by '$(D \vee M)$'. The connective is called DISJUNCTION. We also say that 'D' and 'M' are the DISJUNCTS of the disjunction '$(D \vee M)$'.

Sentence 19 is only slightly more complicated. There are two subjects, but the English sentence only gives the verb once. However, we can paraphrase sentence 19 as 'Either Denison will play golf with me, or Ellery will play golf with me'. Now we can obviously symbolise it by '$(D \vee E)$' again.

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

Sometimes in English, the word 'or' 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, 'Main courses come with chips or mashed potato': you may have chips; you may have mashed potato; but, if you want *both* chips *and* mashed potato, 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. I might play golf with Denison, with Ellery, or with both Denison and Ellery. Sentence 19 merely says that I will play with *at least* one of them. This is called an INCLUSIVE 'OR'. The TFL symbol '$\vee$' always symbolises an inclusive 'or'. (But don't worry, in a moment we will find another way of capturing the exclusive 'or' in TFL.)

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

20. Either you will not have chips, or you will not have mashed potato.
21. You will have neither chips nor mashed potato.
22. You will have either chips or mashed potato, but not both.

We can paraphrase 20 as 'Either *it is not the case that* you will have chips, or *it is not the case that* you have mashed potato'. To symbolise this in TFL, we need both disjunction and negation. Using the symbolisation key from earlier, we can symbolise 'It is not the case that you will have chips' as '$\neg P_1$'. 'It is not the case that you will have mashed potato' is then symbolised as '$\neg P_2$'. So sentence 20 itself is symbolised as '$(\neg P_1 \vee \neg P_2)$'.

Sentence 21 also requires negation. It can be paraphrased as, '*It is not the case that* either you will have chips or you will have mashed potato'. Since this negates the entire disjunction, we symbolise sentence 21 with '$\neg(P_1 \vee P_2)$'.

Sentence 22 is an exclusive 'or'. We can break the sentence into two parts. The first part says that you will have chips or mashed potato. We symbolise this as '$(P_1 \vee P_2)$'. The second part says that you will not have both. We can paraphrase this as: 'It is not the case both that you will have chips and that you will have mashed potato'. Using both negation and conjunction, we symbolise this with '$\neg(P_1 \wedge P_2)$'. Now we just need to put the two parts together. As we saw above, 'but' can usually be symbolised with '$\wedge$'. Sentence 22 can thus be symbolised as '$((P_1 \vee P_2) \wedge \neg(P_1 \wedge P_2))$'.

This last example shows something important. Although the TFL symbol '$\vee$' always symbolises inclusive 'or', we can symbolise 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 Lukas is in Paris, then Lukas is in France.
24. Lukas is in France only if Lukas is in Paris.

Let's use the following symbolisation key:

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

Sentence 23 is roughly of this form: 'if P, then F'. We will use the symbol '$\rightarrow$' to symbolise this 'if…, then…' structure. So we symbolise 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 symbolise this in the same way as sentence 23. That would be a mistake. We all know that sentence 23 must be true (you can't be in Paris without being in France!), but sentence 24 isn't so straightforward: if Lukas were in Dieppe, Lyons, or Toulouse, Lukas would be in France without being in Paris, and sentence 24 would be false. Since geography alone dictates the truth of sentence 23, whereas you need to know Lukas' travel plans to know whether sentence 24 is true, these sentences must mean different things.

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

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

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

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

If we think really hard, all four of these sentences mean the same as 'If Lukas is in Paris, then Lukas is in France'. So they can all be symbolised 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 symbolise English conditionals. We shall return to this in §§9.3 and 11.5.

5.5 Biconditional

Consider these sentences:

29. Shergar is a horse only if he is a mammal
30. Shergar is a horse if he is a mammal
31. Shergar is a horse if and only if he is a mammal

We shall use the following symbolisation key:

H: Shergar is a horse
M: Shergar is a mammal

We already know how to symbolise the first two sentences. Sentence 29 is the easiest: '$H \rightarrow M$'. Sentence 30, on the other hand, can be paraphrased as 'If Shergar is a mammal then Shergar is a horse', and so it can be symbolised as '$M \rightarrow H$'.

But sentence 31 is new. It says something stronger than either 29 or 30. It can be paraphrased as 'Shergar is a horse if he is a mammal, and Shergar is a horse only if Shergar is a mammal'. This is just the conjunction of sentences 29 and 30. So we can symbolise it as '$(H \rightarrow M) \wedge (M \rightarrow H)$'. 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. However, it will prove handy to have a special symbol for the biconditional, and so we will use '$\leftrightarrow$'. So we can symbolise sentence 31 with the TFL sentence '$H \leftrightarrow M$'.

The expression 'if and only if' occurs a lot in philosophy and logic. For brevity, lots of writers abbreviate it with the snappier word 'iff'. (That's not a typo!) 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 symbolised as $\mathcal{A} \leftrightarrow \mathcal{B}$ if it can be paraphrased in English 'A if and only if B', or as 'A iff 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 pudding'. Suppose you ate your greens, but that your parents refused to give you any pudding, on the grounds that they were only committed to the *conditional* (roughly 'if you get pudding, then you will have eaten your greens'), rather than the biconditional (roughly, 'you get pudding 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 symbolise many kinds of sentences. In fact, there is a long list of exercises for you to try at the end of this chapter, which will give you a sense of just how versatile TFL is. But before you get started on all that fun, we should take a moment to talk about a tricky case: the English connective 'unless'.

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

These two sentences are clearly equivalent. To symbolise them, we shall use the symbolisation key:

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

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

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

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

All three are correct symbolisations. Indeed, in chapter 3 we shall see that all three symbolisations are equivalent in TFL.

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

Again, though, there is a little complication. 'Unless' can be symbolised as a conditional; but as I said above, people often use the conditional (on its own) when they mean to use the biconditional. Equally, 'unless' can be symbolised 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 I say: 'I shall go running unless it rains'. I probably mean something like 'I shall go running iff it does not rain' (i.e. the biconditional), or 'either I shall 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, unless you want to be deliberately ambiguous.

Practice exercises

A. Using the symbolisation key given, symbolise 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 symbolisation key given, symbolise 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 symbolisation key given, symbolise 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. Give a symbolisation key and symbolise 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.

E. Give a symbolisation key and symbolise 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.

F. For each argument, write a symbolisation key and symbolise 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.

G. We symbolised an exclusive 'or' using '$\vee$', '$\wedge$', and '$\neg$'. How could you symbolise an exclusive 'or' using only two connectives? Is there any way to symbolise an exclusive 'or' using only one connective?

Sentences of TFL 6

In this chapter, we will provide a *formal definition* of what counts as a sentence of TFL. This is one of the big differences between TFL and natural languages like English: there is no formal definition of an English sentence; we just know them when we see them. But we can be much more precise about TFL.

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. 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 $\mathcal{A}$ of TFL, $\neg\mathcal{A}$ is a sentence of TFL. (Why the funny fonts? I return to this in §7.)

We can say similar things for each of the other connectives. For instance, if $\mathcal{A}$ and $\mathcal{B}$ are sentences of TFL, then $(\mathcal{A} \wedge \mathcal{B})$ 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 $\mathcal{A}$ is a sentence, then $\neg\mathcal{A}$ is a sentence.
3. If $\mathcal{A}$ and $\mathcal{B}$ are sentences, then $(\mathcal{A} \wedge \mathcal{B})$ is a sentence.
4. If $\mathcal{A}$ and $\mathcal{B}$ are sentences, then $(\mathcal{A} \vee \mathcal{B})$ is a sentence.
5. If $\mathcal{A}$ and $\mathcal{B}$ are sentences, then $(\mathcal{A} \rightarrow \mathcal{B})$ is a sentence.
6. If $\mathcal{A}$ and $\mathcal{B}$ are sentences, then $(\mathcal{A} \leftrightarrow \mathcal{B})$ 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. If you want to figure out if an expression is a TFL sentence, you just need to run the recursive definition in reverse: you use the rules to break the expression down into simpler and simpler parts; if you get down to some atomic sentences at the end of this process, then what you started with was a sentence.

Let's consider some examples.

Suppose we want to know whether or not '$\neg\neg\neg D$' is a sentence of TFL. Looking at clause 2 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 clause 2, '$\neg\neg D$' is a sentence *if* '$\neg D$' is. And once again, '$\neg D$' 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 clause 1 of the definition. So '$\neg\neg\neg D$' is a sentence of TFL.

Next, consider the example '$(P \wedge \neg(\neg Q \vee R))$'. This time we need to look at clause 3 of the definition, which tells us that 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 clause 4 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 the MAIN LOGICAL OPERATOR of the sentence. You can find the main logical operator in a sentence by using the following two-step method.

- **Step 1.** Check if the first symbol in the sentence is '$\neg$'; if so, then that '$\neg$' is the main logical connective.
- **Step 2.** If '$\neg$' is not the first symbol, then start counting brackets. Open-brackets '(' are worth $+1$, close-brackets ')' are worth -1. The first connective you hit which isn't a '$\neg$' when your count is at exactly 1 is the main logical connective.

Let's look at a couple of examples. The first symbol in '$\neg\neg\neg D$' is a '$\neg$', so it is the main connective in this sentence. '$(\neg(\neg E \vee F) \to \neg\neg G)$', on the other hand, doesn't start with a '$\neg$', so we need to count brackets: '$(^1\neg(^2\neg E \vee F)^1 \to \neg\neg G)^0$'. The first symbol we hit when the count is at 1 is the first '$\neg$', but the rule is that we ignore negations when we're counting brackets. The next connective we hit when the count is back at 1 is '$\to$', so that is the main connective.

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 chapter 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. (We say that one sentence is a SUBSENTENCE of another iff the former is a part of the latter.) Take the following sentence:

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

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

> The SCOPE of 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![1]

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 shall allow ourselves to use square brackets, '[' and ']', instead of rounded ones. So there is no logical difference between '$(P \lor Q)$' and '$[P \lor Q]$', for example.

Combining these two conventions, we can rewrite the unwieldy sentence

$$(((H \to I) \lor (I \to H)) \land (J \lor K))$$

rather more simply as follows:

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

The scope of each connective is now much clearer.

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

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

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

[1] If you are trying to identify the main connective in a sentence without outermost brackets, you need to tweak the procedure we gave in §6.2. First, you should skip Step 1, because if the outermost brackets have been deleted, then negations can't be the main connective. So, jump straight to Step 2, but now you are looking for the first connective (other than negation) you reach when the bracket-count is 0.

What should you do if you're trying to identify the main connective, but you aren't sure whether the outermost brackets have been included or not? First, try assuming that the outermost brackets have *not* been included. If you can find a connective other than negation when the bracket-count is 0, then that's the main connective. But if you can't, then that means that the sentence does include its outermost brackets, and so you should go back to the original two-step method.

Use and mention 7

In this chapter, we have talked a lot *about* sentences. Now we need to pause on an important, and very general, point.

7.1 Quotation conventions

Consider these two sentences:

- Rishi Sunak is the Prime Minister.
- The expression 'Rishi Sunak' is composed of two uppercase letters and eight 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. And we do so 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 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:

- 'Rishi Sunak' is the Prime Minister.

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

- Rishi Sunak is composed of two uppercase letters and eight lowercase letters.

also says something false: Rishi Sunak is a human, made of flesh and blood, not letters. One final example:

- ' "Rishi Sunak" ' is the name of 'Rishi Sunak'.

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.

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.

We are using 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 no 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 Swash-fonts, quotation marks, and concatenation

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 swash-font letters for this purpose, which look like this:

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

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 $\mathcal{A}$ is a sentence, then $\neg\mathcal{A}$ is a sentence.

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

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

we would not have given a general rule about how to negate sentences in TFL; we would just have given a rule about how to negate the atomic sentence 'A' in particular, and so we would have said nothing about how to negate any other sentence of TFL.

Here is the crucial point:

> '$\mathcal{A}$' is a symbol in augmented English, which we use to talk about any TFL expression. 'A' is a particular atomic sentence of TFL.

But now we face another question. Look again at clause 2 of our recursive definition of a TFL sentence. You should notice that we did not put any quotation marks around '$\neg\mathcal{A}$'. Should we have?

On the one hand, you might think we should. After all, '$\neg\mathcal{A}$' contains '$\neg$', and that isn't English. So we might try to write:

3′. If $\mathcal{A}$ is a sentence, then '$\neg\mathcal{A}$' is a sentence.

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

What we really want to say is something like this:

3″. If $\mathcal{A}$ is a sentence, then the result of writing the symbol '$\neg$' in front of the sentence $\mathcal{A}$ is also a sentence.

This is impeccable, but rather long-winded. However, we can avoid long-windedness by creating our own conventions. We can perfectly well stipulate that an expression like '$\neg\mathcal{A}$' should simply be read as an abbreviation for:

> the result of writing the symbol '$\neg$' in front of the sentence $\mathcal{A}$

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

7.4 Quotation conventions for arguments

TFL is meant to be a tool for studying arguments. So we need some way of representing arguments in TFL. Let's return to one of our earlier examples of an argument:

1. Sharon is an archaeologist.
2. If Sharon is an archaeologist, then Sharon tells Rob a lot about old pots.
3. So: Sharon tells Rob a lot about old pots.

We now know how to symbolise the premises and the conclusion of this argument (using the obvious symbolisation key):

1′. A
2′. $A \to B$
3′. B

However, we do not yet have a way of indicating in TFL which sentences are the premises, and which is the conclusion. So we need a bit more notation. Suppose we want to represent an argument in TFL which has $\mathcal{A}_1, \ldots, \mathcal{A}_n$ as its premises, and $\mathcal{C}$ as its conclusion. Then we will write:

$$\mathcal{A}_1, \ldots, \mathcal{A}_n \therefore \mathcal{C}$$

The role of '$\therefore$' is just to indicate which sentences are the premises, and which is the conclusion. So we can now symbolise the argument 1–3 like this:

1′. A
2′. $A \to B$
3′. $\therefore B$

Chapter 3

Truth-tables

Characteristic truth-tables 8

There are two kinds of sentences in TFL. First we have *atomic* sentences, which are our basic building blocks. We then build *compound* (i.e. non-atomic) sentences by combining the atomic sentences with sentential connectives. Importantly, the truth-value of a compound sentence is always determined by the truth-values of the atomic sentences that make it up.

The easiest way to explain what this means is with some *characteristic truth-tables*. These truth-tables display how each connective maps between truth-values. (For convenience, we shall abbreviate 'True' with 'T' and 'False' with 'F'.)

Negation. For any sentence $\mathcal{A}$: if $\mathcal{A}$ is true, then $\neg\mathcal{A}$ is false; if $\neg\mathcal{A}$ is true, then $\mathcal{A}$ is false. We can summarize this with the characteristic truth-table for negation:

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

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

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

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

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

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

Like conjunction, disjunction is symmetrical.

Conditional. Let's be honest right from the start. Conditionals are a right old mess in TFL. Exactly how much of a mess they are is a matter of *philosophical* contention. We will discuss a few of the subtleties in §9.3. For now, though, let's just stipulate that we're interested in the MATERIAL CONDITIONAL, which has the following truth-conditions: $\mathcal{A} \rightarrow \mathcal{B}$ is false if and only if $\mathcal{A}$ is true and $\mathcal{B}$ is false; otherwise, $\mathcal{A} \rightarrow \mathcal{B}$ is true. We can summarize this with a characteristic truth-table for the material conditional.

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

The conditional is *asymmetrical*. You cannot swap the antecedent and consequent without changing the meaning of the sentence, because $\mathcal{A} \rightarrow \mathcal{B}$ has a very different truth-table from $\mathcal{B} \rightarrow \mathcal{A}$.

Biconditional. Since a biconditional is to be the same as the conjunction of a conditional running in each direction, this is what the truth-table for the biconditional should be:

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

We now need to 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.

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. Consider these two sentences:

1. It is necessarily the case that $2 + 2 = 4$
2. It is necessarily the case that Arthur Conan Doyle wrote sixty *Sherlock Holmes* adventures.

Sentence 1 is true: there is no possible world where $2+2$ is anything other than 4. But sentence 2 is false: there are sad worlds where Arthur Conan Doyle never got the idea to write even one story about a consulting detective. However, note that both sentences are the result of adding the connective 'It is necessarily the case that. . . ' to a true sentence. So, although 'It is necessarily the case that. . . ' is a connective of English, it is not a *truth-functional* connective.

9.2 Symbolising 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 symbolise 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. Sharon is funny and Sharon is kind
2. Although Sharon is funny, Sharon is kind
3. Despite being funny, Sharon is kind
4. Sharon is kind, albeit funny
5. Sharon's funniness notwithstanding, she is kind

All of the above sentences will be symbolised with the same TFL sentence, perhaps '$F \wedge K$'.

In this textbook, we keep talking about using TFL sentences to *symbolise* English sentences. Many other textbooks talk about *translating* English sentences into TFL. But a good translation should preserve meaning, and—as we have just pointed out—there are many ways in which TFL simply cannot do that. This is why we shall speak of *symbolising* English sentences, rather than of *translating* them.

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

F: Sharon is funny.
K: Sharon is kind.

Other textbooks will understand this as a stipulation that the TFL sentence 'F' should *mean* that Sharon is funny, and that the TFL sentence 'K' should *mean* that Sharon is kind. But that is an overstatement of what is going on. Really, the preceding symbolisation key is doing no more nor less than stipulating that the TFL sentence 'F' should take the same truth-value as the English sentence 'Sharon is funny' (whatever that might be), and that the TFL sentence 'K' should take the same truth-value as the English sentence 'Sharon is kind' (whatever that might be).

> When we treat a TFL sentence as *symbolising* an English sentence, we are stipulating that the TFL sentence is to take the same truth-value as that English sentence.

9.3 Troubles with the conditional

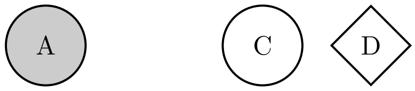
We can bring home the point that TFL *only* deals with truth-functions by considering the case of the conditional. When we introduced the characteristic truth-table for the material conditional in §8, we did not say anything to justify it. Let's now look at one justification, which follows Dorothy Edgington.[1]

Suppose that Matilde has drawn some shapes on a piece of paper, and coloured some of them in. Bruno has not seen them, but he claims:

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

As it happens, Matilde has drawn the following:

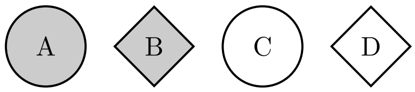
[1] Dorothy Edgington, 2006, 'Conditionals', in the *Stanford Encyclopedia of Philosophy* (http://plato.stanford.edu/entries/conditionals/).



In this case, Bruno's claim is surely true. Shapes C and D are not grey, and so can hardly present *counterexamples* to his claim. Shape A *is* grey, but fortunately it is also circular. So Bruno's claim has no counterexamples. It must be true. And that means that each of the following *instances* of Bruno's 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, now imagine that Matilde draws a fourth shape, like this:



Now Bruno's claim is false. So it must be that this claim is false:

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

Recall that every connective of TFL has to be truth-functional. This means that the mere truth-value 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 '$\rightarrow$' is the *only* 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? Take a look at these English conditionals:

3. If Germany is in Europe, then Berlin is in Europe.
4. If Germany is in Europe, then Paris is in Europe.

We can surely all agree that sentence 3 is obviously true: Berlin is in Germany, so if Germany is in Europe, then Berlin must be in Europe too. But what about sentence 4? It's probably a safe bet that lots of people would say that it's false: Paris isn't in Germany, and so whether Germany is in Europe has nothing to do with whether Paris in Europe! However, something surprising happens when we symbolise these sentences in TFL. Let's use the following symbolisation key:

G: Germany is in Europe.
B: Berlin is in Europe.
P: Paris is in Europe.

We can now symbolise sentences 3 and 4 like this:

3′ $G \rightarrow B$
4′ $G \rightarrow P$

The good news is that sentence 3′ is true, just as it should be. But the unexpected news is that 4′ is *also* true. That's because, according to the truth-table for '$\rightarrow$', '$G \rightarrow P$' is automatically true if it has a true consequent. And 'P' is true on our symbolisation key: Paris *is* in Europe.

As a general rule, then, $\mathscr{A} \rightarrow \mathscr{B}$ is true whenever $\mathscr{B}$ is true, but that isn't how English conditionals seem to work: there seem to be English conditionals, like sentence 4, which are false, even though they have a true consequent.

And that isn't the only way in which our TFL conditional, '$\rightarrow$', seems to work differently from English conditionals. According to its characteristic truth-table, $\mathscr{A} \rightarrow \mathscr{B}$ is also true whenever $\mathscr{A}$ is false. But now look at these two English conditionals:

5. If Germany is in Asia, then Berlin is in Asia.
6. If Germany is in Asia, then Paris is in Asia.

Again, it seems intuitive to say that sentence 5 is true, and sentence 6 is false. But they *both* have false antecedents.

Philosophers disagree about what these cases show us. Some philosophers think that it shows that '$\rightarrow$' is a poor surrogate for conditionals in natural languages like English. Other philosophers—most notably H.P. Grice[2]—argue that all they really show is that our intuitive judgments about which sentences are true and which are false are not always reliable: if we are reluctant to assert a sentence, we often take that to be evidence that the sentence is false; but there can be reasons to refuse to assert a sentence, even if it is true.

However, we do not need to legislate this big philosophical issue here. For now, we can content ourselves with the observation that '$\rightarrow$' is the only candidate for a truth-functional conditional, but that many English conditionals do not appear to be adequately represented by '$\rightarrow$'. TFL is an intrinsically limited language.[3]

[2] H.P. Grice, 1975, 'Logic and conversation', reprinted in: H.P. Grice, 1989, *Studies in the Way of Words*, Cambridge, MA: Harvard University Press.

[3] A note for philosophy of language fans: Things are actually even more complicated than we have made it sound so far, because English contains two different conditionals: an *indicative* conditional, and a *subjunctive* conditional. The following scenario should make the difference clear. Imagine that someone stole Simon's last biscuit, and that Simon accused Daniel of the theft. Now consider these two English conditionals:

7. If Daniel did not steal the biscuit, then Simon wrongly accused him.
8. If Daniel had not stolen the biscuit, then Simon would have wrongly accused him.

Sentence 7 is an indicative conditional, and it is obviously true in our scenario. Sentence 8, on the other hand, is a subjunctive conditional, and it is not obviously true in our scenario. Imagine that Daniel did in fact steal Simon's biscuit, and that no one else had any designs on it. In that case, sentence 8 may well be false: it might be that if Daniel hadn't stolen the biscuit, no one would have, and so Simon wouldn't have accused anyone.

In the main text, we have focussed exclusively on indicative conditionals. We did that because *everyone* agrees that '$\rightarrow$' is a very poor surrogate for the subjunctive conditional.

10 Truth-tables for compound sentences

So far, we have been using symbolisation keys to assign truth-values to atomic TFL sentences, like this:

> B: Big Ben is in London.

But notice that this symbolisation key only fixes the truth-value of 'B' *indirectly*. It tells you that 'B' is to have the same truth-value as the English sentence 'Big Ben is in London', *whatever truth-value that is*. If you want to figure out which truth-value 'B' actually has, you need to go and check whether Big Ben is actually in London.

However, we can also assign truth-values *directly*, if we want. That is, 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. And the truth-table provides us with a means to calculate the truth-value of complex sentences, on each possible valuation. This is easiest to explain by example.

10.1 A worked example

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

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

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

H	I	$(H$	$\vee$	$I)$	$\to$	H
T	T	T		T		T
T	F	T		F		T
F	T	F		T		F
F	F	F		F		F

Now consider the subsentence '$(H \vee I)$'. This is a disjunction, $(\mathcal{A} \vee \mathcal{B})$, with '$H$' as $\mathcal{A}$ and with 'I' as $\mathcal{B}$. The characteristic truth-table for disjunction gives the truth-conditions for *any* sentence of the form $(\mathcal{A} \vee \mathcal{B})$, whatever $\mathcal{A}$ and $\mathcal{B}$ might be. It summarises the point that a disjunction is false only when both of its disjuncts are false. In this case, our disjuncts are just 'H' and 'I'. They are both false on, and only on, the last line of the truth-table. Accordingly, we can calculate the truth-value of the disjunction on all four rows.

		$\mathcal{A}$	$\vee$	$\mathcal{B}$		
H	I	$(H$	$\vee$	$I)$	$\to$	H
T	T	T	T	T		T
T	F	T	T	F		T
F	T	F	T	T		F
F	F	F	F	F		F

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

			$\mathcal{A}$		$\to$	$\mathcal{B}$
H	I	$(H$	$\vee$	$I)$	$\to$	H
T	T		T		T	T
T	F		T		T	T
F	T		T		F	F
F	F		F		T	F

The conditional is the main logical connective of the sentence. The column of 'T's and 'F's underneath the conditional displays the *truth-conditions* for the sentence. In other words, it tells us when the sentence is true, and when it is false: '$(H \vee I) \to H$' is true when 'H' and 'I' are both true, when 'H' is true and 'I' is false, and when 'H' and 'I' are both false; '$(H \vee I) \to H$' is false only when 'H' is false and 'I' is true.

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	$(H$	$\vee$	$I)$	$\to$	H
T	T	T	T	T	**T**	T
T	F	T	T	F	**T**	T
F	T	F	T	T	**F**	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. In this example, we have singled out this column by putting it in bold. When you work through truth-tables yourself, you should also single out the column under the main logical operator (perhaps by writing the column in a different colour, or maybe by drawing a little arrow pointing it out).

10.2 Building truth-tables

Truth-tables give us a way of surveying all the different ways of assigning True and False to a stock of atomic sentences. Earlier we called each of these ways a *valuation*, and each valuation is represented by a line on the truth-table.

The size of a 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 '$[(C \leftrightarrow C) \rightarrow \neg C] \vee \neg(C \wedge C)$'. The truth-table for this sentence 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	$[(C\leftrightarrow C)\rightarrow\neg C]$	$\vee$	$\neg(C\wedge C)$
T	T T T F F T	**F**	F T T T
F	F T F T T F	**T**	T F F F

Looking at the column underneath the main logical operator, we see that the sentence is false when 'C' is true, and true when 'C' is false. In other words, the very complex sentence '$[(C \leftrightarrow C) \rightarrow \neg C] \vee \neg(C \wedge C)$' has exactly the same truth-table as the much simpler sentence '$\neg C$'.

If we are dealing with a sentence that contains two atomic sentences, then we'll need a truth-table with four lines. We've already looked quite closely at an example like that, back when we were studying the truth-table for '$(H \vee I) \rightarrow H$'.

A sentence that contains three atomic sentences requires a truth-table with 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 can see how the truth-value of '$M \wedge (N \vee P)$' is determined by the truth-values of 'M', 'N', and 'P'.

A 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 truth-table has n different atomic sentences, then it must have 2^n lines.

In order to fill in the columns of a truth-table, begin with the right-most 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 bracketing conventions

Consider these two sentences:

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

These have the same truth-table. 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). And since the order of the brackets does not matter, we can allow ourselves to drop them. In short, we can save some ink and some eyestrain by writing:

$$A \wedge B \wedge C$$

The general point is that, if we just have a long list of conjunctions, we can drop the inner brackets. (We already said we could drop outermost brackets in §6.) The same observation holds for disjunctions. Since the following sentences have exactly the same truth-table:

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

we can simply write:

$$A \vee B \vee C$$

And generally, if we just have a long list of disjunctions, we can drop the inner brackets. *But be careful.* These two sentences have *different* truth-tables:

$$((A \to B) \to C)$$
$$(A \to (B \to C))$$

So if we were to write:

$$A \to B \to C$$

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: you can drop brackets when dealing with a long list of conjunctions, or when dealing with a long list of disjunctions. But that's it.

Practice exercises

A. Present 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

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 shall introduce some related ideas, and show how to use truth-tables to test whether or not they apply.

11.1 Tautologies and tautological contradictions

In §3, we introduced *necessary truth* and *necessary falsity*. Both notions have surrogates in TFL. We shall start with a surrogate for necessary truth.

> $\mathcal{A}$ 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. Here is an example:

X	Y	Z	$[X \vee (Y \wedge Z)] \rightarrow (X \vee Y)$
T	T	T	T T T T T **T** T T T
T	T	F	T T T F F **T** T T T
T	F	T	T T F F T **T** T T F
T	F	F	T T F F F **T** T T F
F	T	T	F T T T T **T** F T T
F	T	F	F F T F F **T** F T T
F	F	T	F F F F T **T** F F F
F	F	F	F F F F F **T** F F F

It is important to emphasise that being a tautology really is just a *surrogate* for necessary truth. There are some necessary truths that we cannot adequately symbolise in TFL. An example is '$2 + 2 = 4$'. This *must* be true, but if we try to symbolise it in TFL, the best we can offer is an atomic sentence, and no atomic sentence is a tautology. Still, if we can adequately symbolise 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:

> $\mathcal{A}$ is a TAUTOLOGICAL CONTRADICTION iff it is false on every valuation.

We can determine whether a sentence is a tautological 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 tautological contradiction. Here is an example:

A	B	$(A \leftrightarrow B) \wedge \neg(B \rightarrow A)$
T	T	T T T **F** F T T T
T	F	T F F **F** T T F F
F	T	F F T **F** F F T T
F	F	F T F **F** F F T F

11.2 Tautological equivalence

Here is another useful notion:

> $\mathcal{A}$ and $\mathcal{B}$ are TAUTOLOGICALLY EQUIVALENT iff they have the same truth-value on every valuation.

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 tautologically equivalent. Again, it is easy to test for tautological equivalence using truth-tables. Consider the sentences '$\neg(P \vee Q)$' and '$\neg P \wedge \neg Q$'. Are they tautologically equivalent? We can find out by drawing up a truth-table, except this time we need to evaluate two sentences at once:

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 tautologically equivalent.

11.3 Tautological consistency

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

> $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are JOINTLY TAUTOLOGICALLY CONSISTENT iff there is some valuation which makes them all true.

Derivatively, sentences are JOINTLY TAUTOLOGICALLY INCONSISTENT iff there is no valuation that makes them all true. Again, it is easy to test for joint tautological consistency using truth-tables.

11.4 Tautological entailment and tautological validity

We now come to the key notion of this chapter:

> The sentences $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ TAUTOLOGICALLY ENTAIL the sentence $\mathcal{C}$ iff there is no valuation of the atomic sentences which makes all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ true and $\mathcal{C}$ false.
>
> We also call an argument TAUTOLOGICALLY VALID iff its premises tautologically entail its conclusion.

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

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

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

11.5 Validity: tautological and original recipe

How does this new idea of *tautological validity* relate to the original idea of *validity* that we first introduced back in §2? (Recall that an argument is valid iff it is impossible for all of its premises to be true and its conclusion false.) Well, there's some good news!

> If an argument is tautologically valid, then it is valid.

Here's why. If $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ is tautologically valid, then $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ tautologically entail $\mathcal{C}$. In other words, there is no valuation which makes all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ true whilst making $\mathcal{C}$ false. This means that it is *logically impossible* for $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ all to be true whilst $\mathcal{C}$ is false. But this is just what it takes for $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ to be valid!

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

This is an important milestone: a test for the validity of arguments! But we should not get carried away just yet, because there's also some bad news: although every tautologically valid argument is valid, not every valid argument is tautologically valid. Take a look at this example:

Huey has four legs.
So: Huey has more than two legs.

This argument is clearly valid. More than that, it is clearly *formally valid.* Here is another argument with the same valid form:

Rob ate four cookies.
So: Rob ate more than two cookies.

We might display the valid form that these arguments share like this:

a bears relation R to four Fs.
So: a bears relation R to more than two Fs.

But unfortunately, TFL doesn't have the resources to handle forms like that. If we wanted to symbolise an argument of this form in TFL, we would have no choice but just to use two different atomic sentences—perhaps 'F' and 'T'—for the premise and the conclusion. And it is obvious that no atomic sentence can ever tautologically entail another atomic sentence.

This point reveals an important *limit* on what we have achieved so far. Although we have a test for validity, we do not yet have a test for *invalidity*. If an argument passes our test—i.e. if it is tautologically valid—then it follows that the argument is valid. But if an argument fails the test—i.e. if it is not tautologically valid—then it does not automatically follow that the argument is invalid. (Or to put it in terms that everyone in the post-COVID world can understand: this test for validity has no false positives, but plenty of false negatives.)

Here's another limit to beware of. We can only apply our test for validity once we have symbolised an argument in TFL. But it isn't always obvious how to do that. Consider this English argument:

Jane is not unhappy.
So: Jane is happy.

How should we symbolise this argument in TFL? The obvious suggestion would be something like this:

$$\neg\neg H \therefore H$$

This TFL argument is tautologically valid: a short truth-table will quickly show that the premise tautologically entails the conclusion. But the original English argument itself doesn't seem valid: in English, it sounds OK to say that Jane is neither happy nor unhappy, but just indifferent.

Now, to be clear, this example does not show that there is anything wrong with our test for validity. The reason that we got into trouble is that 'Jane is not unhappy' should not be symbolised as '$\neg\neg H$'. But this does illustrate a general point. We need to be very careful when we're applying our test for validity. Before we can apply the test to an argument written in a natural language like English, we need to symbolise it in TFL. And if we get the symbolisation wrong, then the test won't tell us anything about the argument that we started with.

11.6 The double-turnstile

We are going to use the notion of tautological entailment rather a lot in this module. It will help us, then, to introduce a symbol that abbreviates it:

> $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{C}$ iff $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ tautologically entail $\mathcal{C}$.
>
> Equivalently: $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{C}$ iff $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ is a tautologically valid argument.

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

But we should 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 '$P \to Q$' tautologically 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 a special limiting case:

$\vDash \mathcal{C}$ iff $\mathcal{C}$ is a tautology.

It might initially look a bit weird not to write any sentences on the left of the double turnstile, but it does make sense. We are still saying that there is no valuation which makes all of the sentences on the left of the double turnstile true, whilst making $\mathcal{C}$ false. However, since there are *no* sentences on the left of this double turnstile, this just means that there is no valuation which makes $\mathcal{C}$ false. Otherwise put, it says that every valuation makes $\mathcal{C}$ true. And that is precisely to say that $\mathcal{C}$ is a tautology.

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

Let's take a moment to compare and contrast '$\vDash$' with '$\to$'.

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

Now note: $\vDash \mathcal{A} \to \mathcal{C}$ (i.e. $\mathcal{A} \to \mathcal{C}$ is a tautology) iff there is no valuation of the atomic sentences that makes $\mathcal{A} \to \mathcal{C}$ false. Since a conditional is true except when its antecedent is true and its consequent false, $\vDash \mathcal{A} \to \mathcal{C}$ iff there is no valuation that makes $\mathcal{A}$ true and $\mathcal{C}$ false.

Combining these two observations, we see that $\vDash \mathcal{A} \to \mathcal{C}$ iff $\mathcal{A} \vDash \mathcal{C}$. 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.

Practice exercises

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

B. Use truth-tables to determine whether these sentences are jointly tautologically consistent, or jointly tautologically 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. $A \leftrightarrow (B \vee C)$, $C \to \neg A$, $A \to \neg B$

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

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

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

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

E. Consider the following principle:

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

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 will look at some shortcuts to help you along your 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	$(P \lor Q)$	$\leftrightarrow$	$\neg P$
T	T	T	**F**	F
T	F	T	**F**	F
F	T	T	**T**	T
F	F	F	**F**	T

You also know for sure that a disjunction is true whenever one of the disjuncts is true. So if you find a true disjunct, there is no need to work out the truth-values of the other disjuncts. Here's an example of this shortcut in action:

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

Equally, you know for sure that a conjunction is false whenever one of the conjuncts is false. So if you find a false conjunct, there is no need to work out the truth-value of the other conjunct. Here's an example:

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

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

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 tautological validity and tautological entailment

When we use truth-tables to test for tautological 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 tautological validity:

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

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

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

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

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

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

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

This truth-table is enough to show that the argument is tautologically valid. But 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!

12.3 Shortcuts in other tests

We can take similar shortcuts in our other truth-table tests. If you want to show that a sentence *is* a tautology, you need to show that it is true on *every* line of its truth-table. But if you want to show that a sentence is *not* a tautology, you only need to show that it is false on at least *one* line. So this partially completed truth-table is enough to show that '$B \vee (B \wedge \neg A)$' is not a tautology:

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

In exactly the same way, to show that a sentence is a contradiction, you need to show that it is false on every line of its truth-table; but to show that a sentence is not a contradiction, you only need to show that it is true on at least one line.

Now imagine that you want to show that a pair of sentences are tautologically equivalent. That would require showing that they have the same truth-value on every line of their truth-tables. But to show that they are *not* tautologically equivalent, you just need to show that they have different truth-values on at least one line. Take a look at this partial truth-table:

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

We have only filled in the first line, but it is already enough to show that '$A \vee B$' is not tautologically equivalent to '$\neg A \leftrightarrow B$':

Last, but not least, we come to tests for tautological consistency and inconsistency. If you want to show that a collection of sentences are jointly tautologically *consistent*, you only need to show that there is at least one line where they are all true together. So this partial truth-table demonstrates that '$P \vee Q$', '$\neg P \vee Q$' and '$\neg P \vee \neg Q$' are jointly tautologically consistent:

P	Q	$P \vee Q$	$\neg P \vee Q$	$\neg P \vee \neg Q$
T	T			
T	F			
F	T	**T**	**T**	T **T**
F	F			

If, on the other hand, you want to show that a collection of sentences are jointly tautologically *inconsistent*, you need to show that there is no line on their truth-tables where they are all true together. That means you can't skip any lines. However, you can give up on a line as soon as you find that a sentence in the collection is false. So this partial truth-table shows that '$P \leftrightarrow Q$', '$\neg P$' and '$\neg\neg Q$' are jointly tautologically inconsistent:

P	Q	$P \leftrightarrow Q$	$\neg P$	$\neg\neg Q$
T	T		**F**	
T	F			**F** T
F	T	**F**		
F	F			**F** T

This table handily summarises when you need to go through every line of the truth-table, and when just one line will do:

	Yes	**No**
tautology?	all lines	one line
tautological contradiction?	all lines	one line
tautologically equivalent?	all lines	one line
tautologically consistent?	one line	all lines
tautologically valid?	all lines	one line

Practice exercises

A. Determine whether each sentence is a tautology, a tautological contradiction, or neither. Feel free to use shortcuts, if you would like!

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. $(B \wedge D) \leftrightarrow [A \leftrightarrow (A \vee C)]$

B. Determine whether these pairs of sentences are tautologically equivalent. Feel free to use shortcuts, if you would like!

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

C. Determine whether these sentences are jointly tautologically consistent, or jointly tautologically inconsistent. Feel free to use shortcuts, if you would like!

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

D. Determine whether these arguments are tautologically valid. Feel free to use shortcuts, if you would like!

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

Chapter 4

Natural deduction for TFL

13 The very idea of natural deduction

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

While we have been working in TFL, we used truth-tables to sharpen this idea up. 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.

But 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. And it might be nice to keep track of these different forms of inference.

The aim of this chapter is to introduce you to a new way of testing arguments for validity. You will learn how to construct a *formal proof* that a conclusion follows from some premises. These proofs are very demanding, since you only have access to a very limited stock of inference rules. But there is a real benefit to tying one arm behind your back like this. The inference rules you will use are *obviously* good rules, and so if they let you derive a conclusion from some premises, you know that that conclusion really follows from those premises.

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:

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

If you wanted to use a truth-table to check if this argument is tautologically valid, your truth-table would need to be 1024 lines long. I suppose you *could* do that (and if you did, you would find out that this argument *is* tautologically valid), but now consider this argument:

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

This argument is also tautologically valid, but showing that with a truth-table would require $2^{20} = 1048576$ lines! In principle, we could set a machine to grind through truth-tables and report back when it is finished. In practice, complicated arguments in TFL can become *impracticable* if we use truth-tables.

And things will only get worse later on. In Chapter 5, we will step up from TFL to *First-Order Logic* (FOL). This is a huge upgrade: FOL is *much* more powerful than TFL. But, as always, there's a catch. There's just nothing like a truth-table test for FOL. The only way to show that an argument is valid in FOL is with a proof. The good news is that the proof-system for FOL is an extension of the proof-system for TFL. So learning how to write proofs for TFL is an investment that will really pay off by the end of the textbook!

There are several different ways of setting up proof-systems. In this textbook, we will work with a *natural deducion* system. The modern development of natural deduction dates from simultaneous and unrelated papers by Gerhard Gentzen and Stanisław Jaśkowski (1934). However, the natural deduction system that we shall consider is based largely around work by Frederic Fitch (first published in 1952).

Basic rules for TFL 14

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

14.1 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 just call them '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 shall start a proof by writing the premise:

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

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

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. (More on that shortly!) We are hoping to conclude that '$\neg A \wedge \neg B$'; so we are hoping ultimately to conclude our proof with

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

for some number n. It doesn't matter how long the proof is—a thousand line proof is still a proof!—but we would obviously prefer a shorter proof to a longer one.

Similarly, suppose we wanted to provide a proof for this argument:

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

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.

14.2 Conjunction

Suppose I want to show that logic is both fun and useful. One obvious way to do this would be as follows: first I show that logic is fun; then I show that logic is useful; then I put these two demonstrations together, to obtain the conjunction.

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

F: Logic is fun
U: Logic is useful

Perhaps you are working through a proof, and you have obtained 'F' on line 8 and 'U' on line 15. Then on any subsequent line you can infer '$F \wedge U$', like this:

8	F	
15	U	
	$F \wedge U$	$\wedge$I 8, 15

Note that whenever we infer a new line in our proof, we must justify that inference by some rule. We cite '$\wedge$I 8, 15' here to indicate that the line is justified by the rule of conjunction introduction ($\wedge$I) applied to lines 8 and 15. You could equally well obtain:

8	F	
15	U	
	$U \wedge F$	$\wedge$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 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 on. But when we define the rule, we use variables to emphasise 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 logic is both fun and useful. You are entitled to conclude that logic is fun. Equally, you are entitled to conclude that logic is useful. 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 infer either of the conjuncts by $\wedge$E. But it is important to emphasise this point: you can only apply this rule when conjunction is the main logical operator. So you cannot infer 'D' just from '$C \vee (D \wedge E)$'! This point generalises: *you cannot apply an operator's elimination rule to a sentence, unless that operator is the main operator in that sentence.* (Equally, when you introduce an operator, you always introduce it as the *main* operator.)

Even with just these two rules, we can start to see some of the power of our formal proof system. Here's something we all know about conjunctions: you can validly swap the order of the conjuncts. So, for example, the following argument must be valid:

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

We can use our new natural deduction system to provide a proof for this argument. We begin by writing down the premise, which is our assumption. We indicate that this is the premise of our argument by drawing a line underneath it. So the beginning of the proof looks like this:

$$\begin{array}{r|l}
1 & [(A \vee B) \to (C \vee D)] \wedge [(E \vee F) \to (G \vee H)] \\ \cline{2-2}
\end{array}$$

Our aim is to find a way of getting from our premise to our desired conclusion by repeatedly applying our rules of proof. The main operator in the premise is '$\wedge$', so we can use $\wedge$E to infer each conjunct, like this:

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

We can now apply $\wedge$I to lines 3 and 2 (in that order) to infer the desired conclusion. The finished proof looks like this:

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

This is a very simple proof, but it shows how we can chain rules of proof together into longer proofs. In passing, note that investigating this argument with a truth-table would have required 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 vindicating this argument, we start by writing:

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

From the premise, we can get each of the conjuncts by applying $\wedge$E twice. At this point, our proof looks like this:

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

Line 3 is another conjunction, so we can apply $\wedge$E twice more:

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

And now we can reintroduce conjunctions in the order we want them. So 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. When we discussed semantics, we became a bit more relaxed, and allowed ourselves to drop inner brackets in long conjunctions, since the order of the brackets did not affect the truth-table. The proof just given suggests that we could also drop these inner brackets in our proofs. However, that is not standard, and we will not do it. Instead, we shall return to the more austere bracketing conventions. (Though we will still allow ourselves to drop outermost brackets, for legibility.)

Let's look at one final illustration. When using the $\wedge$I rule, there is no need to apply it to different sentences. So we can formally prove 'A' from 'A' as follows:

1	A	
2	$A \wedge A$	$\wedge$I 1, 1
3	A	$\wedge$E 2

Simple, but effective.

14.3 The conditional

Consider the following argument:

> If Sharon is an archaeologist, then she wishes that she were Indiana Jones. Sharon is an archaeologist. So Sharon wishes that she were Indiana Jones.

This argument is certainly valid. And it suggests a straightforward CONDITIONAL ELIMINATION rule ($\rightarrow$E):

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

This rule is also sometimes called *modus ponens*. Again, this is an elimination rule, because it allows us to infer a sentence that may not contain '$\rightarrow$', having started with a sentence that did contain '$\rightarrow$'. 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 $\rightarrow$E, we always cite the conditional first, followed by the antecedent.

Let's look at an example of conditional elimination in action. Consider the following valid argument:

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

As always, we start by writing out our premises. There are two of them this time, but we only underline the final premise:

$$\begin{array}{r|l} 1 & A \rightarrow (B \wedge C) \\ 2 & \underline{A} \end{array}$$

Line 1 is a conditional, and line 2 is the antecedent to that conditional. So we can now apply $\rightarrow$E:

$$\begin{array}{r|ll} 1 & A \rightarrow (B \wedge C) & \\ 2 & \underline{A} & \\ 3 & B \wedge C & \rightarrow\text{E } 1, 2 \end{array}$$

And now we can bring in $\wedge$E to infer the conclusion we were looking for:

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

Note that we couldn't apply $\wedge$E to line 1, because conjunction is not the main operator in that line. The main operator in line 1 is the conditional, and so that is the only operator we can eliminate at that point.

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

> Sharon is an archaeologist. Therefore, if Sharon reads Latin, then she is an archaeologist and she reads Latin.

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

> Assume that Sharon is an archaeologist. Now, *additionally* assume, just for the sake of argument, that Sharon reads Latin. Then by conjunction introduction—which we just discussed—Sharon is an archaeologist *and* she reads Latin. Of course, that's conditional on the extra assumption that Sharon reads Latin. But this just means that, *if* Sharon reads Latin, *then* she is an archaeologist and she reads Latin.

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

$$\begin{array}{r|l} 1 & \underline{A} \end{array}$$

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

$$\begin{array}{r|l} 1 & \underline{A} \\ 2 & \quad \vert\ \underline{L} \end{array}$$

Note that we are *not* claiming, on line 2, to have inferred 'L' from line 1. So we do not need to write in any justification for the additional assumption on line 2. (In our natural deduction system, you never need to justify your assumptions!) 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 $\wedge$I. So we could continue our proof:

$$\begin{array}{r|ll} 1 & \underline{A} & \\ 2 & \quad \vert\ \underline{L} & \\ 3 & \quad \vert\ A \wedge L & \wedge\text{I } 1, 2 \end{array}$$

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

$$\begin{array}{r|ll} 1 & \underline{A} & \\ 2 & \quad \vert\ \underline{L} & \\ 3 & \quad \vert\ A \wedge L & \wedge\text{I } 1, 2 \\ 4 & L \to (A \wedge L) & \to\text{I } 2\text{–}3 \end{array}$$

Observe that we have dropped back to using one vertical line. We have DISCHARGED the additional assumption, 'L', since the conditional itself follows just from our original assumption, 'A'. This is very important to emphasise: when we discharge an assumption, that assumption does not hang around as a premise of our argument; it was a temporary assumption made during the course of the argument, but the argument as a whole does not rely upon it.

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:

i	$\quad \mathcal{A}$	
j	$\quad \mathcal{B}$	
	$\mathcal{A} \to \mathcal{B}$	$\to$I i–j

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 $\to$I in action. Suppose we want to give a proof to vindicate the following argument:

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

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

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

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

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

14.4 Additional assumptions and subproofs

The rule $\to$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	$\quad B \wedge B$	$\wedge$I 2, 2
4	$\quad B$	$\wedge$E 3
5	$B \to B$	$\to$I 2–4

This is perfectly in keeping with the rules we have laid down already. And it should not seem particularly strange. Since '$B \to B$' 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	$\quad B \wedge B$	$\wedge$I 2, 2
4	$\quad B$	$\wedge$E 3
5	$B \to B$	$\to$I 2–4
6	B	naughty attempt to invoke $\to$E 5, 4

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. But if you tell me that Sharon is an archaeologist (symbolised by 'A'), I shouldn't be able to conclude that Queen Boudica stood twenty-feet tall (symbolised by 'B')! So 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 made 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 inferred from our original assumptions. (After all, those original assumptions are still in effect.) But at some point, we will want to stop working with the additional assumption: we shall 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 naughty 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 the same as DISCHARGING the assumption that initiated the 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	$\mid \underline{A}$	
2	$\mid \quad \mid \underline{B}$	
3	$\mid \quad \mid \quad \mid \underline{C}$	
4	$\mid \quad \mid \quad \mid A \wedge B$	$\wedge$I 1, 2
5	$\mid \quad \mid C \to (A \wedge B)$	$\to$I 3–4
6	$\mid B \to (C \to (A \wedge B))$	$\to$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. For 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 I tell you that Sharon is an archaeologist, you should not be able to infer that, if Sharon likes cakes (symbolised by 'C') then *both* Sharon is an archaeologist 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. And you never need to include any justification for a new assumption. However, it is always best to have a plan when you pick a new assumption. Starting a subproof with an arbitrary, wacky assumption would just waste lines of the proof. But if you wanted to prove a conditional with $\to$I, then that rule tells you that you need to 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.

14.5 The biconditional

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

In order to prove '$W \leftrightarrow X$', for instance, you must be able to prove 'X' on the assumption 'W' *and* prove 'W' on the assumption 'X'. The BICONDITIONAL INTRODUCTION rule ($\leftrightarrow$I) therefore requires two subproofs:

$$\begin{array}{l|l}
i & \quad \mathcal{A} \\
j & \quad \mathcal{B} \\
k & \quad \mathcal{B} \\
l & \quad \mathcal{A} \\
 & \mathcal{A} \leftrightarrow \mathcal{B} \qquad \leftrightarrow\text{I } i\text{–}j,\ k\text{–}l
\end{array}$$

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

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

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

and equally:

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

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

14.6 Disjunction

Suppose that Sharon is an archaeologist. Then either Sharon is either an archaeologist or a physicist. After all, to say that Sharon is either an archaeologist or a physicist is to say something weaker than to say that she is an archaeologist.

We should emphasise this point. Suppose that Sharon is an archaeologist. It follows that Sharon is *either* an archaeologist *or* a kumquat. Equally, it follows that *either* Sharon is an archaeologist *or* the Moon is a hollow megastructure. Generally speaking, it follows that *either* Sharon is an archaeologist *or* [INSERT ANY SENTENCE]. Any inference of this shape would be valid.

(Some of them might be a little strange, but there would be nothing wrong with them *logically* speaking.)

Armed with all this, we can present the DISJUNCTION INTRODUCTION rule(s):

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

and

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

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

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

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

The disjunction elimination rule is, though, slightly trickier. Suppose that Sharon is either an archaeologist or a physicist. What can you conclude? Not that Sharon is an archaeologist; it might be that she is a physicist instead. And equally, not that Sharon is a physicist; it might turn out that she is an archaeologist after all. Disjunctions, just by themselves, are hard to work with.

But we *could* reason like this:

> Assume that Sharon is either an archaeologist or a physicist. If Sharon is an archaeologist, then she must be very smart. But, equally, if Sharon is a physicist, then she must be very smart. So whichever one Sharon is—archaeologist or physicist—we can safely conclude that she must be very smart.

This pattern of reasoning can be expressed in this DISJUNCTION ELIMINATION ($\vee$E) rule:

$$
\begin{array}{r|ll}
m & \mathcal{A} \vee \mathcal{B} & \\
i & \quad \begin{array}{|l} \mathcal{A} \\ \hline \end{array} & \\
j & \quad \begin{array}{|l} \mathcal{C} \end{array} & \\
k & \quad \begin{array}{|l} \mathcal{B} \\ \hline \end{array} & \\
l & \quad \begin{array}{|l} \mathcal{C} \end{array} & \\
 & \mathcal{C} & \vee\text{E } m,\ i\text{–}j,\ k\text{–}l
\end{array}
$$

This obviously looks a bit clunkier than our previous rules, but the point is fairly simple. Suppose we have some disjunction, $\mathcal{A} \vee \mathcal{B}$. Suppose we have two subproofs, showing us that $\mathcal{C}$ follows from the assumption that $\mathcal{A}$, and that $\mathcal{C}$ follows from the assumption that $\mathcal{B}$. Then we can infer $\mathcal{C}$ 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 devise 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. And that just involves checking every line, and making sure that it is justified in accordance with the rules we have laid down.

14.7 Contradiction

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

Imagine that you're arguing with someone, and you're trying to convince them that they're wrong. If you're *very* lucky, you might manage to convince your opponent that they are contradicting themselves. At that point, you'll have them on the ropes. They'll have no choice but 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 'But that's absurd!' And the CONTRADICTION INTRODUCTION rule is that we can infer '$\bot$' whenever we find an explicit contradiction, i.e. whenever we find both a sentence and its negation appearing in our proof:

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

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

Our CONTRADICTION ELIMINATION '$\bot$' is just a formalisation of the *explosion* rule from §3.5. Recall that absolutely everything, no matter how absurd, follows from a contradiction (and so contradictions blow formal systems up). Here is the formal rule:

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

Note that $\mathcal{A}$ can be *any* sentence whatsoever: it could be an atom, but it could also be as complex of a sentence as you like!

A final remark. We 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, to flag that a line of reasoning involves a contradiction. (The symbol '$\bot$' would then be 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 shall *officially* plump for the second. '⊥' is to be read as abbreviating some canonical contradiction. This means that we can manipulate it, in our proofs, just like any other sentence.

14.8 Negation

There is obviously a tight link between contradiction and negation. Indeed, the ⊥I rule essentially behaves as a rule for negation elimination: we introduce '⊥' 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 NEGATION INTRODUCTION rule:

i	$\quad \mathcal{A}$	
j	$\quad \bot$	
	$\neg\mathcal{A}$	$\neg$I i–j

There can be as many lines between i and j as you like. Let's look at an example of this rule in action. Consider the following valid argument:

$$P \to (Q \land R), \neg R \therefore \neg P$$

Here's a proof that uses ¬I to vindicate this argument:

1	$P \to (Q \land R)$	
2	$\neg R$	
3	$\quad P$	
4	$\quad Q \land R$	$\to$E 1, 3
5	$\quad R$	$\land$E 4
6	$\quad \bot$	$\bot$I 5, 2
7	$\neg P$	$\neg$I 3–6

We also need to add an extra rule for negation. It is a lot like the rule used in disjunction elimination, and it requires a little motivation.

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

This line of thinking motivates the following rule:

i	$\quad\mathcal{A}$	
j	$\quad\mathcal{B}$	
k	$\quad\neg\mathcal{A}$	
l	$\quad\mathcal{B}$	
	$\mathcal{B}$	TND i–j, k–l

The rule is sometimes called TERTIUM NON DATUR, which means 'no third way'. There can be as many lines as you like between i and j, and as many lines as you like between k and l. Moreover, the subproofs can come in any order, and the second subproof does not need to come immediately after the first.

To see the rule in action, consider:

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

Here is a proof corresponding with the argument:

1	P	
2	$\quad D$	
3	$\quad P \wedge D$	$\wedge$I 1, 2
4	$\quad (P \wedge D) \vee (P \wedge \neg D)$	$\vee$I 3
5	$\quad \neg D$	
6	$\quad P \wedge \neg D$	$\wedge$I 1, 5
7	$\quad (P \wedge D) \vee (P \wedge \neg D)$	$\vee$I 6
8	$(P \wedge D) \vee (P \wedge \neg D)$	TND 2–4, 5–7

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

Practice exercises

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

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

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.

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

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

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

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

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

Additional rules for TFL 15

In §14, we introduced the basic rules of our proof system for TFL. In this section, we shall add some additional rules to our system. These will make our system much easier to work with. (However, in §17 we will see that they are not strictly speaking *necessary*.)

15.1 Reiteration

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

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

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

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

This is a fairly typical use of the R rule.

15.2 Disjunctive syllogism

Here is a very natural argument form.

> Ahmed is either in York or in London. He is not in London. So, he is in York.

This inference pattern is called DISJUNCTIVE SYLLOGISM. We add it to our proof system as follows:

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

and

m	$\mathcal{A} \lor \mathcal{B}$	
n	$\neg\mathcal{B}$	
	$\mathcal{A}$	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. (This is, if you like, a new rule of disjunction elimination.)

15.3 Modus tollens

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

> If Sofia is running late, then she will have skipped breakfast. Sofia has not skipped breakfast. So Sofia is not running late.

This inference pattern is called MODUS TOLLENS. The corresponding rule is:

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

As usual, the premises may occur in either order, but we always cite the conditional first. (This is, if you like, a new rule of conditional elimination.)

15.4 Double-negation elimination

Another useful rule is DOUBLE-NEGATION ELIMINATION. This rule does exactly what it says on the tin:

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

Here's a justification for this inference: if $\neg\neg\mathcal{A}$ is true, then $\neg\mathcal{A}$ is false; if $\neg\mathcal{A}$ is false, then $\mathcal{A}$ is true; so if $\neg\neg\mathcal{A}$ is true, then $\mathcal{A}$ is true.

However, as we noted at the end of §11.5, we need to be careful when applying this kind of reasoning to natural language sentences. 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.

15.5 De Morgan Laws

Our final additional rules are called DE MORGAN'S LAWS. (These are named after the 19th Century logician, August De Morgan.) The shape of the rules should be familiar from truth-tables.

The first De Morgan rule is:

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

The second De Morgan is the reverse of the first:

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

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

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

And the fourth is the reverse of the third:

m	$\neg\mathcal{A} \wedge \neg\mathcal{B}$	
	$\neg(\mathcal{A} \vee \mathcal{B})$	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:

$$\begin{array}{r|l}
1 & W \to \neg B \\
2 & A \wedge W \\
3 & B \vee (J \wedge K) \\
\hline
4 & W \\
5 & \neg B \\
6 & J \wedge K \\
7 & K
\end{array}$$

$$\begin{array}{r|l}
1 & L \leftrightarrow \neg O \\
2 & L \vee \neg O \\
\hline
3 & \quad \mid\ \neg L \\
 & \quad \mid\ \overline{} \\
4 & \quad \mid\ \neg O \\
5 & \quad \mid\ L \\
6 & \quad \mid\ \bot \\
7 & \neg\neg L \\
8 & L
\end{array}$$

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

B. Give a proof for each of these arguments:

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

Proof strategies 16

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 $\mathcal{A} \rightarrow \mathcal{B}$, plan to use the $\rightarrow$I rule. This requires starting a subproof in which you assume $\mathcal{A}$. The subproof ought to end with $\mathcal{B}$. So, what can you do to get $\mathcal{B}$?

Work forwards from what you have. When you are starting a proof, look at the premises; later, look at the sentences that you have 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 using TND. If all else fails, trying using TND. Alternatively, you could try working indirectly. If you are trying to prove $\mathcal{A}$, then you could start by assuming $\neg\mathcal{A}$. If a contradiction follows, then you will be able to obtain $\neg\neg\mathcal{A}$ by $\neg$I, and then $\mathcal{A}$ by DNE.

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

Derived rules 17

When laying out the natural deduction system for TFL, we separated the rules into two batches: first we introduced the *basic* rules (§14), and then we added some helpful, quality-of-life-improving *extra* rules (§15). In this section, we will show that these extra rules are not strictly speaking necessary, but can be derived from the basic rules.

17.1 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 §15. But equally well, you can do this with the *basic* rules of §14:

m	$\mathscr{A}$	
k	$\mathscr{A} \wedge \mathscr{A}$	$\wedge$I 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 which can be proven using the rule R, can be proven (with one more line) using just the *basic* rules of §14. So we can describe the rule R as a DERIVED rule, since its justification is derived from our basic rules.

17.2 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 $\mathcal{B}$. You can do this with the rule of DS, introduced in §15. But equally well, you can do this with the *basic* rules of §14:

$$
\begin{array}{l|lll}
m & \mathcal{A} \vee \mathcal{B} & & \\
n & \neg\mathcal{A} & & \\
k & \quad| \; \underline{\mathcal{A}} & & \\
k+1 & \quad| \; \bot & & \bot\text{I } k,\, n \\
k+2 & \quad| \; \mathcal{B} & & \bot\text{E } k+1 \\
k+3 & \quad| \; \underline{\mathcal{B}} & & \\
k+4 & \quad| \; \mathcal{B} \wedge \mathcal{B} & & \wedge\text{I } k+3,\, k+3 \\
k+5 & \quad| \; \mathcal{B} & & \wedge\text{E } k+4 \\
k+6 & \mathcal{B} & & \vee\text{E } m,\, k\text{–}k+2,\, k+3\text{–}k+5
\end{array}
$$

So the DS rule, again, can be derived from our more basic rules. Adding it to our system did not make any new proofs possible. Anytime you use the DS rule, you could always take a few extra lines and prove the same thing using only our basic rules. It is a *derived* rule.

17.3 Derivation of Modus tollens

Suppose you have the following in your proof:

$$
\begin{array}{l|l}
m & \mathcal{A} \to \mathcal{B} \\
n & \neg\mathcal{B}
\end{array}
$$

You now want, on line k, to prove $\neg\mathcal{A}$. You can do this with the rule of MT, introduced in §15. But equally well, you can do this with the *basic* rules of §14:

$$
\begin{array}{l|lll}
m & \mathcal{A} \to \mathcal{B} & & \\
n & \neg\mathcal{B} & & \\
k & \quad| \; \underline{\mathcal{A}} & & \\
k+1 & \quad| \; \mathcal{B} & & \to\text{E } m,\, k \\
k+2 & \quad| \; \bot & & \bot\text{I } k+1,\, n \\
k+3 & \neg\mathcal{A} & & \neg\text{I } k\text{–}k+2
\end{array}
$$

Again, the rule of MT can be derived from the *basic* rules of §14.

17.4 Derivation of double-negation elimination

Consider the following deduction scheme:

m	$\neg\neg\mathcal{A}$	
k	$\quad \mathcal{A}$	
$k+1$	$\quad \mathcal{A}$	R k
$k+2$	$\quad \neg\mathcal{A}$	
$k+3$	$\quad \bot$	$\bot$I $k+2$, m
$k+4$	$\quad \mathcal{A}$	$\bot$E $k+3$
$k+5$	$\mathcal{A}$	TND k–$k+1$, $k+2$–$k+4$

Again, then, we can derive the DNE rule from the *basic* rules of §14.

17.5 Derivation of De Morgan rules

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

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

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

m	$\neg\mathcal{A} \vee \neg\mathcal{B}$	
k	$\mathcal{A} \wedge \mathcal{B}$	
$k+1$	$\mathcal{A}$	$\wedge$E k
$k+2$	$\mathcal{B}$	$\wedge$E k
$k+3$	$\neg\mathcal{A}$	
$k+4$	$\bot$	$\bot$I $k+1$, $k+3$
$k+5$	$\neg\mathcal{B}$	
$k+6$	$\bot$	$\bot$I $k+2$, $k+5$
$k+7$	$\bot$	$\vee$E m, $k+3$–$k+4$, $k+5$–$k+6$
$k+8$	$\neg(\mathcal{A} \wedge \mathcal{B})$	$\neg$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 §§15–18 used derived rules. Replace the use of derived rules, in such proofs, with only basic rules. You will find some 'repetition' in the resulting proofs; in such cases, offer a streamlined proof using only basic rules. (This will give you a sense, both of the power of derived rules, and of how all the rules interact.)

Proof-theoretic concepts 18

We shall introduce some new vocabulary:

$\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{C}$ iff $\mathcal{C}$ can be proven from $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ (i.e. there is a proof which starts with $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ and ends with $\mathcal{C}$, without relying on any additional undischarged assumptions).

The symbol '$\vdash$' is called the SINGLE TURNSTILE. It is important to emphasise that this is not the *double turnstile* symbol ('$\vDash$') that we used to symbolise tautological validity in chapter 3 (and validity in FOL in chapter 6). 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.*

To show that a conclusion is provable from some premises, you just have to come up with a single suitable proof. It is typically much harder to show that a conclusion is *not* provable from some premises. To do this, you would have to demonstrate, not just that certain proof strategies fail, but that *no* proof of the conclusion from the premises is possible. No amount of failed attempts will demonstrate that: perhaps there is a proof out there, but it's just too long and complex for you to make out.

Now that we have this new '$\vdash$' symbol, we can push our notation a bit further:

$\mathcal{A}$ is a THEOREM iff $\vdash \mathcal{A}$

In other words, a sentence is a theorem iff we can prove it without relying on *any* premises. To illustrate this, suppose that you want to prove that the following is a theorem:

$$\neg(A \wedge \neg A)$$

To show that this sentence is a theorem, you need to come up with a proof for it that doesn't appeal to any premises. However, that doesn't mean that you can't make any assumptions at all. You can, so long as all of your assumptions are *discharged* by the end of your proof. You're currently trying to prove a negation, so you will need to kick-off a subproof by assuming '$A \wedge \neg A$'. Your aim is to show that this assumption leads to a contradiction, and then use negation introduction to discharge that assumption and get to the conclusion you're after in the main proof. When you're all finished, your proof might look like this:

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

This proof doesn't rely on any premises (i.e. undischarged assumptions), and so it shows that '$\neg(A \wedge \neg A)$' is a theorem. This particular theorem is an instance of what is sometimes called *the Law of Non-Contradiction*.

Again, showing that a sentence *is* a theorem is much easier than showing that a sentence *is not* a theorem: it takes just one proof to show that a sentence is a theorem; but to show that a sentence is not a theorem, you would have to show that there is no possible way to prove the sentence without relying on premises.

Here is another new bit of terminology:

> Two sentences $\mathcal{A}$ and $\mathcal{B}$ are **provably equivalent** iff each can be proved from the other; i.e., both $\mathcal{A} \vdash \mathcal{B}$ and $\mathcal{B} \vdash \mathcal{A}$.

Yet again, 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: you need to show that it is impossible to prove one of the sentences from the other.

Here is a third, related, bit of terminology:

> The sentences $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are **jointly contrary** iff a contradiction can be proved from them (i.e. $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \bot$).

One last time: It is easy to show that some sentences are jointly contrary; you just need to prove a contradiction from assuming all the sentences. Showing that some sentences are not jointly contrary is much harder. You would have to show that a contradiction cannot be proven from those premises.

This table summarises whether one or two proofs suffice, or whether we must reason about all possible proofs.

	Yes	**No**
provable?	one proof	all possible proofs
theorem?	one proof	all possible proofs
equivalent?	two proofs	all possible proofs
contrary?	one proof	all possible proofs

Practice exercises

A. Show that each of the following sentences is a theorem:

1. $O \to O$
2. $N \lor \neg N$
3. $J \leftrightarrow [J \lor (L \land \neg L)]$
4. $((A \to B) \to A) \to A$

B. Provide proofs to show each of the following:

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

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. $T \to S$, $\neg S \to \neg T$
4. $U \to I$, $\neg(U \land \neg I)$
5. $\neg(C \to D), C \land \neg D$
6. $\neg G \leftrightarrow H$, $\neg(G \leftrightarrow H)$

D. If you know that $\mathcal{A} \vdash \mathcal{C}$, what can you say about $(\mathcal{A} \land \mathcal{B}) \vdash \mathcal{C}$? What about $(\mathcal{A} \lor \mathcal{B}) \vdash \mathcal{C}$? Explain your answers.

E. According to a result known as the DEDUCTION THEOREM, $\mathcal{A} \vdash \mathcal{C}$ iff $\vdash \mathcal{A} \to \mathcal{C}$. Give a demonstration of this result.

Chapter 5

First-Order Logic

Building blocks of FOL 19

19.1 The need to decompose sentences

Consider the following argument:

> Ruth is a logician. All logicians are great dinner guests. So Ruth is a great dinner guest.

This argument is clearly valid. More than that, it is clearly valid *in virtue of its form*. However, if we try to symbolise it in TFL, the best we can do is offer a symbolisation key like this:

> L: Ruth is a logician.
> A: All logicians are great dinner guests.
> D: Ruth is a great dinner guest.

And the argument itself becomes:

$$L, A \therefore D$$

The premises of this argument do not *tautologically* entail its conclusion, and so the argument is not *tautologically* valid. But the original English argument is clearly *valid*: it is obviously impossible for its premises to be true and its conclusion to be false.

The problem is not that we have made a mistake in the symbolisation the argument. This is the best symbolisation we can give *in TFL*. The problem lies with TFL itself. 'All logicians are great dinner guests' is about the relation between being a logician and table manners. But we lose all that structure when we symbolise it with the blank atomic sentence 'A', and we can no longer see the connection between Ruth's being a logician and her being a great dinner guest.

We are hitting up against a hard limit of TFL. The basic units of TFL are atomic sentences, and TFL cannot decompose them any further. To symbolise arguments like the preceding one, we will have to develop a new logical language which will allow us to *split the logical 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 logical atom.

First, we have *names*. In FOL, we indicate these with lowercase italic letters. For instance, we might let 'r' stand for Ruth Barcan-Marcus, or let 'a' stand for Arthur Prior.

Second, we have predicates. English predicates are expressions with gaps, like '_____ is a logician' or '_____ is a philosopher'. 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 'Ruth is a logician' or 'Arthur is a philosopher'. In FOL, we indicate predicates with uppercase italic letters. For instance, we might let the FOL predicate 'L' symbolise the English predicate '_____ is a logician'. Then the expression 'Lr' will be a sentence in FOL, which symbolises the English sentence 'Ruth Barcan-Marcus is a logician'. Equally, we might let the FOL predicate 'P' symbolise the English predicate '_____ is a philosopher'. Then the expression 'Pa' will symbolise the English sentence 'Arthur Prior is a philosopher'.

Third, we have quantifiers. For instance, '$\exists$' will roughly convey 'there is at least one ...'. So we might symbolise the English sentence 'there is a philosopher' with the FOL sentence '$\exists x Px$', which we could read out-loud as 'there is at least one thing, x, such that x is a philosopher'.

That is the general idea. But FOL is significantly more subtle than TFL. So we will come at it slowly.

19.2 Names

In English, a *singular term* is a word or phrase that refers to a *specific* person, place, or thing. For example, I have a dog called 'Munnery', and so 'Munnery' is a singular term that refers to my dog. By contrast, the word 'dog' is *not* a singular term, because its job is not to refer to any particular person, place, or thing; its job is to categorise things in a certain way.

There are two different kinds of singular term, and they pick out individuals in different ways. Some singular terms pick out individuals by description. Take the phrase 'the first dog that Rob adopted', for example. This term picks out whichever dog I happened to adopt first. As it turns out, this singular term picks out Munnery, but if things had gone differently, and I had adopted some other dog first, then it would have picked out that other dog instead. We call singular terms like this *definite descriptions*.[1]

We can contrast definite descriptions with *proper names*. These are expression that pick out individuals without describing them in any way. You can think of a proper name as just a handy tag attached to a particular individual. We have already seen one example of a proper name, 'Munnery': this term picks out one of my dogs, but you couldn't guess anything about Munnery just from his name. I have another dog called 'Kitson', and 'Kitson' is another proper name: again, you can't tell anything about Kitson just by looking at his name. Or, if you would like some examples that don't have anything to do with my pets, here are some familiar proper names from pop culture: 'Tom Cruise', 'Donald Glover', 'Sandra Oh'; you might know a lot about all of these people, but none of it is given away by their names alone.

For now, we will focus on proper names, and set definite descriptions to one side. (But don't worry, we'll come back to definite descriptions in §23!) In FOL, our NAMES are the lower-case letters from 'a' to 'r'. We can add numerical subscripts if we want to use some letter more than once. So here are

[1] *Indefinite descriptions* are expressions like 'a dog that Rob adopted'. Note that indefinite descriptions are clearly not any kind of singular term.

some names 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 important difference. 'Munnery' is the name of one of my dogs, but it is *also* the name of one of the best stand-up comedians in the world.[2] 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 symbolisation keys. These indicate, temporarily, what a name shall pick out. Here's an example of what a symbolisation key for some names might look like:

a: Ayo Edebiri
e: Ebon Moss-Bachrach
j: Jeremy Allen White

19.3 Predicates

The simplest predicates are expressions which attribute properties to individuals. They let us say things about objects. Here are some examples of English predicates:

_____ is an actor
_____ starred in *The Bear*
The 2023 Golden Globe for Best Actor was awarded to _____

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, 'Ayo talked with Ebon about Jeremy'. This sentence contains three different singular terms, and so we can make three predicates by removing them one at a time:

_____ talked with Ebon about Jeremy
Ayo spoke with _____ about Jeremy
Ayo spoke with Ebon about _____

In FOL, we use capital letters as predicates. (We can also add numerical subscripts to the capital letters, just to make sure that we won't run out of predicates.) We might add predicates to our symbolisation key like this:

L: _____ is running late
M: _____ is memorising their lines
a: Ayo Edebiri
e: Ebon Moss-Bachrach
j: Jeremy Allen White

Using this symbolisation key, we can start to symbolise some English sentences that use these names and predicates in combination. For example, consider the English sentences:

[2] https://en.wikipedia.org/wiki/Simon_Munnery

1. Jeremy is running late.
2. Ayo and Ebon are memorising their lines.
3. If Jeremy is running late, then Norman and Danai are memorising their lines.

Sentence 1 is straightforward: we symbolise it by 'Lj'.

Sentence 2: this is a conjunction of two simpler sentences. The simple sentences can be symbolised just by 'Ma' and 'Me'. Then we help ourselves to our resources from TFL, and symbolise the entire sentence by '$Ma \land Me$'. 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 symbolise this with '$Lj \to (Ma \land Me)$'.

19.4 Quantifiers

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

4. Everyone is running late.
5. Someone is memorising their lines.

It might be tempting to symbolise 4 with something like this, '$La \land Le \land Lj$'. However, this would only say that Ayo, Ebon, and Jeremy are running late. There might be some other people who we have not yet introduced names for, and when we say that *everyone* is running late, we mean those people too. In order to express this, we need to introduce a new symbol, '$\forall$'. This is called the UNIVERSAL QUANTIFIER.

A quantifier must always be followed by a variable. In FOL, variables are italic lowercase letters from 's' to 'z' (with or without numerical subscripts). So we might symbolise sentence 4 as '$\forall xLx$'. 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, and then 'Lx' says of the person you picked out that they are running late.

It is important to note that there is no special reason to use 'x' rather than some other variable. The sentences '$\forall xLx$', '$\forall yLy$', '$\forall zLz$', and '$\forall x_5Lx_5$' use different variables, but they will all be logically equivalent.

To symbolise sentence 5, we introduce another new symbol: the EXISTENTIAL QUANTIFIER, '$\exists$'. Like the universal quantifier, the existential quantifier requires a variable. Sentence 5 can be symbolised by '$\exists xMx$'. Whereas '$\forall xMx$' is read naturally as 'for all x, x is memorising their lines', '$\exists xMx$' is read naturally as 'there is something, x, such that x is memorising their lines'. Once again, the variable is a kind of placeholder; we could just as easily have symbolised sentence 5 with '$\exists zMz$', '$\exists w_{256}Mw_{256}$', or whatever.

Some more examples will help. Consider these further sentences:

6. No one is running late.
7. Someone is not memorising their lines.
8. Not everyone is memorising their lines.

Sentence 6 can be paraphrased as, 'It is not the case that someone is running late'. We can then symbolise it using negation and an existential quantifier:

'$\neg\exists xLx$'. But sentence 6 could also be paraphrased as, 'Everyone is not running late'. With this in mind, it can be symbolised using negation and a universal quantifier: '$\forall x\neg Lx$'. Both of these are acceptable symbolisations. Indeed, it will transpire that, in general, $\forall x\neg\mathcal{A}$ is logically equivalent to $\neg\exists x\mathcal{A}$. (Notice that we have here returned to the practice of using '$\mathcal{A}$' as a metavariable, from §7.) Symbolising a sentence one way, rather than the other, might seem more 'natural' in some contexts, but it is not much more than a matter of taste.

Sentence 7 is most naturally paraphrased as, 'There is some x, such that x is not memorising their lines'. This then becomes '$\exists x\neg Mx$'. Of course, we could equally have written '$\neg\forall xMx$', which we would naturally read as 'It is not the case that everyone is memorising their lines'. And that would be a perfectly adequate symbolisation of sentence 8.

19.5 Domains

Given the symbolisation key we have been using, '$\forall xLx$' symbolises 'Everyone is running late'. 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 almost never mean everyone who was ever alive or who will ever live. We normally mean something more modest: all the stars of *The Bear*, or everyone involved in the production of the show, 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 Atlanta, we define the domain to be people in Atlanta. We write this at the beginning of the symbolisation key, like this:

domain: people in Chicago

The quantifiers *range over* the domain. Given this domain, '$\forall x\ldots$' is to be read roughly as 'Every person in Chicago is such that…' and '$\exists x\ldots$' 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 'someone is running late' from 'Ayo is running late'. In FOL, then, we shall want to be able to infer '$\exists xLx$' from 'La'. We can guarantee this by insisting that each name must pick out exactly one thing *in the domain*. If we want to name people in places beside Atlanta, 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.

Sentences with one quantifier 20

We now have all of the basics of FOL in place. Symbolising 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. So after you read all of the *explanations* to follow, make sure you do all of the *exercises*!

20.1 Complex adjectives

Here's a sentence about a magic sentient car from some old Disney movies:

1. Herbie is a white car

We can paraphrase this as 'Herbie is white and Herbie is a car'. We can then use a symbolisation key like:

W: _____ is white
C: _____ is a car
h: Herbie

This allows us to symbolise sentence 1 as '$Wh \land Ch$'. This kind of strategy works in a lot of cases: very often, you can take a sentence attributing a complex adjective to an individual, and break it down into a conjunction. But now consider:

2. Marcus Rashford is a short footballer.
3. Marcus Rashford is a man.
4. Marcus Rashford is a short man.

Following the case of Herbie, we might try to use a symbolisation key like:

S: _____ is short
F: _____ is a footballer
M: _____ is a man
r: Marcus Rashford

Then we would symbolise sentence 2 with '$Sr \land Fr$', sentence 3 with 'Mr' and sentence 4 with '$Sr \land Mr$'. But that would be a terrible mistake! For this now suggests that sentences 2 and 3 together *entail* sentence 4. But they do not. Marcus Rashford is 1.80m tall, which is taller than the average man (1.77m), but shorter than the average footballer (1.82m). The point is that sentence 2 says that Marcus Rashford is short *qua* footballer, even though he is tall *qua* man. So you will need to symbolise '_____ is a short footballer' and '_____ is a short man' using completely different predicates.

Similar examples abound. All chefs are people, but some good chefs are bad people. I am a fairly unreliable story teller, but a trustworthy logic teacher. And so it goes. The moral is: when you see two adjectives in a row, you need to ask yourself carefully whether they can be treated as a conjunction or not.

20.2 Complex quantifier phrases

Consider these sentences:

5. Every coin in my pocket is a copper.
6. Some coin on the table is a silver.
7. Not all the coins on the table are coppers.
8. None of the coins in my pocket are silvers.

In providing a symbolisation 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 need to deal with any names. So here is our key:

domain: all coins
P: _____ is in my pocket
T: _____ is on the table
C: _____ is a copper
S: _____ is a silver

Sentence 5 is most naturally symbolised using a universal quantifier. The universal quantifier says something about everything in the domain, not just about the coins in my pocket. Sentence 5 can be paraphrased as 'for any coin, *if* that coin is in my pocket *then* it is a copper'. So we can symbolise it as '$\forall x(Px \rightarrow Cx)$'.

It is important to notice that we used a conditional here, not a conjunction. That is, we didn't try to symbolise 5 with '$\forall x(Px \land Cx)$': that FOL sentence really symbolise 'every coin is both in my pocket, and a copper'. This obviously means something very different than sentence 5! And so we see:

> A sentence can be symbolised as $\forall x(\mathcal{F}x \rightarrow \mathcal{G}x)$ if it can be paraphrased in English as 'every F is G'.

Sentence 6 is most naturally symbolised using an existential quantifier. It can be paraphrased as 'there is some coin which is both on the table and which is a silver'. So we can symbolise it as '$\exists x(Tx \land Sx)$'.

Notice that, this time, we used a conjunction rather than a conditional. Suppose we had instead written '$\exists x(Tx \rightarrow Sx)$'. That would mean that there is some object in the domain of which '$(Tx \rightarrow Sx)$' is true. Recall that, in TFL, $\mathcal{A} \rightarrow \mathcal{B}$ is tautologically equivalent to $\neg\mathcal{A} \lor \mathcal{B}$. This equivalence will also hold in FOL. So '$\exists x(Tx \rightarrow Sx)$' is true iff there is some object in the domain, such that '$(\neg Tx \lor Sx)$' is true of that object. In plainer words, '$\exists x(Tx \rightarrow Sx)$'

is true iff some coin is *either* not on the table *or* is a silver. 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 '$\exists x(Tx \rightarrow Sx)$' 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 *really* sure that you need one.

> A sentence can be symbolised as $\exists x(\mathcal{F}x \wedge \mathcal{G}x)$ if it can be paraphrased in English as 'some F is G'.

Sentence 7 can be paraphrased as, 'It is not the case that every coin on the table is a copper'. So we can symbolise it by '$\neg\forall x(Tx \rightarrow Cx)$'. You might look at sentence 7 and paraphrase it instead as, 'Some coin on the table is not a copper'. You would then symbolise it by '$\exists x(Tx \wedge \neg Cx)$'. Although it might not be immediately obvious yet, these two sentences are logically equivalent. (This is due to the logical equivalence between $\neg\forall x\mathcal{A}$ and $\exists x\neg\mathcal{A}$, mentioned in §19, along with the equivalence between $\neg(\mathcal{A} \rightarrow \mathcal{B})$ and $\mathcal{A} \wedge \neg\mathcal{B}$.)

Sentence 8 can be paraphrased as, 'It is not the case that there is a silver in my pocket'. This can be symbolised by '$\neg\exists x(Px \wedge Sx)$'. It might also be paraphrased as, 'Everything in my pocket is not a silver', and then could be symbolised by '$\forall x(Px \rightarrow \neg Sx)$'. Again the two symbolisations are logically equivalent. Both are correct symbolisations of sentence 8.

20.3 Empty predicates

In §19, we emphasised 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. It is worth taking a moment to look at how empty predicates behave.

Suppose we want to symbolise these two sentences:

9. Every monkey knows sign language
10. Some monkey knows sign language

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

domain: animals
M: _____ is a monkey
S: _____ knows sign language

Sentence 9 can now be symbolised by '$\forall x(Mx \rightarrow Sx)$'. Sentence 10 can be symbolised as '$\exists x(Mx \wedge Sx)$'.

But now, here's an interesting question: Does sentence 9 *logically entail* sentence 10? You might be tempted at first to say that it does. In other words, you might think that it is impossible for it to be the case that every monkey knows sign language, without it also being the case that some monkey knows sign language. But that 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? Well, suppose that there were no monkeys in our domain. In that case, '$\forall x(Mx \rightarrow Sx)$' would automatically be true. To see why, we again need to remember that $\mathcal{A} \rightarrow \mathcal{B}$ is tautologically equivalent to $\neg\mathcal{A} \vee \mathcal{B}$. So '$\forall x(Mx \rightarrow Sx)$' is true iff '$\neg Mx \vee Sx$' is true of everything in the domain. In plainer words, '$\forall x(Mx \rightarrow Sx)$' is true iff each thing in the domain either knows sign language, or isn't a monkey. Clearly, then, one way to make '$\forall x(Mx \rightarrow Sx)$' true is to not put any monkeys in your domain!

Here, then, is the general rule:

> When $\mathcal{F}$ is an empty predicate, a sentence $\forall x(\mathcal{F}x \rightarrow \ldots)$ will be VACUOUSLY true.

20.4 Picking a domain

The appropriate symbolisation of an English language sentence in FOL will depend on the symbolisation key. Choosing a key can be difficult. Suppose we want to symbolise the English sentence:

11. Every rose has a thorn.

We might begin by writing out this symbolisation key:

R: _____ is a rose
T: _____ has a thorn

It is tempting to say that sentence 11 should be symbolised 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 symbolisation. 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 vacuously true. This is not what we want. To symbolise sentence 11 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 11 can, if we like, be symbolised with '$\forall xTx$'. 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 symbolise 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 symbolise sentences like:

12. Every cowboy sings a sad, sad song.

Our domain must now include both all the roses (so that we can symbolise sentence 11) and all the cowboys (so that we can symbolise sentence 12). So we might offer the following symbolisation key:

domain: people and plants
C: _____ is a cowboy
S: _____ sings a sad, sad song
R: _____ is a rose
T: _____ has a thorn

Now we will have to symbolise sentence 11 with '$\forall x(Rx \rightarrow Tx)$', since '$\forall xTx$' would symbolise the sentence 'every person or plant has a thorn'. Similarly, we will have to symbolise sentence 12 with '$\forall x(Cx \rightarrow Sx)$'.

Here's a useful point to remember here. The universal quantifier can be used to symbolise 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'.

20.5 Helpful paraphrases

When symbolising English sentences in FOL, it is important to understand the structure of the sentences you want to symbolise. What matters is the final symbolisation 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 finally symbolise directly in FOL.

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

domain: people
S: _____ is a singer
P: _____ is a pop star
b: Beyoncé

Now consider these sentences:

13. If Beyoncé is a singer, then she is a pop star.
14. If someone is a singer, then she is a pop star.

The same words appear as the consequent in sentences 13 and 14 ('... she is a pop star'), but they mean very different things. To make this clear, it often helps to paraphrase the original sentences, removing pronouns.

Sentence 13 can be paraphrased as, 'If Beyoncé is a singer, then *Beyoncé* is a pop star'. This can obviously be symbolised as '$Sb \rightarrow Pb$'.

Sentence 14 must be paraphrased differently: 'If a person is a singer, then *that person* is a pop star'. This sentence is not about any particular person, so we need a variable. As a halfway house, we can paraphrase this as, 'For any person x, if x is a singer, then x is a pop star'. Now this can be symbolised as '$\forall x(Sx \rightarrow Px)$'. This is the same sentence we would have used to symbolise 'Everyone who is a singer is a pop star'. And on reflection, that is surely true iff sentence 14 is true, as we would hope.

Consider these further sentences:

15. If anyone is a singer, then Beyoncé is a pop star.
16. If anyone is a singer, then she is a pop star.

The same words appear as the antecedent in sentences 15 and 16 ('If anyone is a singer…'). But it can be tricky to work out how to symbolise these two uses. Again, paraphrase will come to our aid.

Sentence 15 can be paraphrased, 'If there is at least one singer, then Beyoncé is a pop star'. It is now clear that this is a conditional whose antecedent is a quantified expression; so we can symbolise the entire sentence with a conditional as the main logical operator: '$\exists x Sx \rightarrow Pb$'. Note that, in this case, we have ended up symbolising the English quantifier-phrase 'anyone' with the *existential* quantifier, '$\exists$'.

Sentence 16 can be paraphrased, 'For all people x, if x is a singer, then x is a pop star'. Or, in more natural English, it can be paraphrased by 'All singers are pop stars'. It is best symbolised as '$\forall x(Px \rightarrow Sx)$', just like sentence 14. Note that, in this case, we have ended up symbolising the English quantifier-phrase 'anyone' with the *universal* quantifier, '$\forall$'.

The moral is that the English words 'any' and 'anyone' should typically be symbolised using quantifiers, but *which* quantifier depends on context. 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'.

20.6 Quantifiers and scope

Continuing the example, suppose I want to symbolise these sentences:

17. If everyone is a singer, then Rob is a singer
18. Everyone is such that, if she is a singer, then Rob is a singer.

To symbolise these sentences, we need to add a new name to the symbolisation key, namely:

r: Rob Trueman

Sentence 17 is a conditional, whose antecedent is 'everyone is a singer'. So we will symbolise it with '$\forall x Sx \rightarrow Sr$'. This sentence is *necessarily* true: if *everyone* is indeed a singer, then take any one you like—for example, me—and they will be a singer.

Sentence 18, by contrast, might best be paraphrased by 'every person x is such that, if x is a singer, then Rob is a singer'. This is symbolised by '$\forall x(Sx \rightarrow Sr)$'. And this sentence is false. Beyoncé is a definitely a singer. So 'Sb' is true. But I am definitely *not* a singer, and so 'Sr' is false. Accordingly, '$Sb \rightarrow Sr$' will be false. And thus '$\forall x(Sx \rightarrow Sr)$' will be false as well.

In short, '$\forall x Sx \rightarrow Sr$' and '$\forall x(Sx \rightarrow Sr)$' 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 (§5.1), and it will help to explain it in this way.

In the sentence '$\neg Sb \rightarrow Sr$', the scope of '$\neg$' is just the antecedent of the conditional. We are saying something like: if 'Sb' is false, then 'Sr' is true. Similarly, in the sentence '$\forall x Sx \rightarrow Sr$', the scope of '$\forall x$' is just the antecedent of the conditional. We are saying something like: if 'Sx' is true of *everything* in the domain, then 'Sr' is also true.

In the sentence '$\neg(Sb \rightarrow Sr)$', the scope of '$\neg$' is the entire sentence. We are saying something like: '$(Sb \rightarrow Sr)$' is false. Similarly, in the sentence '$\forall x(Sx \rightarrow Sr)$', the scope of '$\forall x$' is the entire sentence. We are saying something like: the conditional '$(Sx \rightarrow Sr)$' is true of *everything* in the domain. (Or equivalently: '$(\neg Sx \lor Sr)$' is true of everything in the domain.)

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.

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.

Symbolise each argument in FOL.

B. Using the following symbolisation key:

domain: people
K: _____ knows the combination to the safe
S: _____ is a spy
V: _____ is a vegetarian
h: Hofthor
i: Ingmar

symbolise 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 symbolisation key:

domain: all animals
A: _____ is an alligator.
M: _____ is a monkey.
R: _____ is a reptile.
Z: _____ lives at the zoo.
a: Amos
b: Bouncer
c: Cleo

symbolise 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 symbolisation key and symbolise 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 generality 21

So far, we have only considered sentences that require one-place predicates and one quantifier. The full power of FOL really comes out when we start to use many-place predicates and multiple quantifiers. For this insight, we largely have Gottlob Frege (1879) to thank, but also Peirce.

21.1 Many-placed predicates

All of the predicates that we have considered so far express properties that objects might have. The predicates have one gap in them, and to make a sentence, we simply need to slot in one name. They are ONE-PLACE predicates.

But other predicates express *relations* 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 names in order to make a sentence. Conversely, if we start with an English sentence containing many names, we can make a two-place predicate by removing two names. Consider the sentence 'York is between London and Edinburgh', for example. We can make three different two-place predicates just by removing two names from this sentence:

York is between _____ and _____
_____ is between London and _____
_____ is between _____ and Edinburgh

And we can make a THREE-PLACE predicate by removing all three names:

_____ is between _____ and _____

Indeed, there is no upper limit on the number of places that our predicates may contain.

21.2 Mind the gap(s)!

Now that we have the idea of predicates that come with more than one gap, we need to figure out how to symbolise them in FOL.

Strictly speaking, we need to introduce some sort of label to tell us how many gaps a predicate comes with. We can do that by adding numerical superscripts to our predicates: one-place predicates get superscript '1's, e.g. 'A^1', 'B^1_2', 'Z^1_{15}'; two-place predicates get superscript '2's, e.g. 'A^2', 'B^2_{247}', 'Z^2_{5000}'; and so on.

We make atomic sentences by combining n-place predicates with n names. For example, 'A^2mn' is an atomic sentence that combines the two-place predicate 'A^2' with two names, 'm' and 'n'. Note that we are allowed to repeat names. For example, 'T^3aba' is an atomic sentence that combines the three-place predicate 'T^3' with the names 'a', 'b', and then 'a' again.

Now, I said that our predicates need superscripts, *strictly speaking*. However, they're pretty ugly, and *most* of the time you can tell how many gaps a predicate has from the way that we use it. If you write a sentence that combines a predicate with five names (e.g. '$Rabcde$'), then, unless you've made a terrible mistake, the predicate must have five gaps! So, for the most part, we can be kind to ourselves, and not bother adding the little superscripts to our predicates. We'll only include them when it actually helps.

Next we need to figure out how to write symbolisation key entries for predicates with many gaps. Suppose you wanted to symbolise these two sentences in FOL:

1. Imre loves Karl
2. Karl loves Imre

You might start by drawing up this symbolisation key:

domain: people
i: Imre
k: Karl
L: _____ loves _____

But what, exactly, does 'Lik' symbolise, according to this key? We might say that it symbolises 'Imre loves Karl', but it is not the only possible answer. Two-place predicates apply to objects *in a particular order*: '_____ loves _____' applies to the *lover* first, and to the *beloved* second; so in 'Imre loves Karl', '_____ loves _____' applies to Imre first, and to Karl second. It is up to us to choose whether we want our FOL predicate 'L' to apply to objects in the same order as '_____ loves _____'. If we decide that it should, then 'Lik' will symbolise sentence 1; but if we instead decide that it should apply in the *opposite* order, then 'Lik' will symbolise sentence 2.

We need to indicate our choice of order in the symbolisation key. We will do this by numbering the blanks in English predicates, like this:

L: $\_\_\_\_\__1$ loves $\_\_\_\_\__2$
M: $\_\_\_\_\__2$ loves $\_\_\_\_\__1$

These numbers indicate the order in which our FOL predicates apply to objects: 'L' applies to the lover first and to the beloved second, whereas 'M' applies to the beloved first and to the lover second. This key now definitively tells us that 'Lik' and 'Mki' both symbolise sentence 1, whereas 'Lki' and 'Mik' both symbolise sentence 2.

So far, we have only looked at cases where we fill the different gaps in a predicate with different names. But it is important to note that we can fill different gaps with the *same* name, if we like. Suppose, for example, that we want to symbolise these sentences in FOL:

3. Imre loves Imre
4. Karl loves Karl

We can symbolise sentence 3 with 'Lii', and sentence 4 with 'Lkk'. Or, equally, we could symbolise them with 'Mii' and 'Mkk'. Or, if we are feeling very fancy, we could introduce a new one-place predicate into our symbolisation key, like this:

N^1: $\underline{\quad\quad}_1$ loves $\underline{\quad\quad}_1$

At first, you might think that 'N' is a two-place predicate, because it looks like it has two gaps. However, by labelling both gaps with a '1', we are stipulating that both gaps must be filled by the same name at the same time, and so we are effectively treating them as *one* gap. We could then symbolise sentence 3 with 'Ni', and sentence 4 with 'Nk'.

All of this carries over to predicates with more than two gaps in the obvious way. So, for example, let's add these entries to our symbolisation key:

P^3: $\underline{\quad\quad}_1$ loves $\underline{\quad\quad}_2$ more than he loves $\underline{\quad\quad}_3$
Q^3: $\underline{\quad\quad}_2$ loves $\underline{\quad\quad}_3$ more than he loves $\underline{\quad\quad}_1$
R^2: $\underline{\quad\quad}_1$ loves $\underline{\quad\quad}_1$ more than he loves $\underline{\quad\quad}_2$

Using this key, we could symbolise 'Imre loves himself more than he loves Karl' with any of the following: '$Piik$', '$Qkii$', and 'Rik'.

21.3 The order of quantifiers

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

5. Each person has someone that they love.
6. There is some particular person who is loved by everyone.

Sentence 5 can be symbolised by '$\forall x \exists y Lxy$', which you can translate back into logician's English as, 'For each person x, there is some person y, such that x loves y'. This sentence would be true of a love-triangle. For example, suppose that our domain of discourse is restricted to Imre, Juan and Karl. Suppose also that Karl loves Imre but not Juan, that Imre loves Juan but not Karl, and that Juan loves Karl but not Imre. Then sentence 5 is true.

Sentence 6 is symbolised by '$\exists y \forall x Lxy$', which you can translate back into logician's English as, 'There is some person y such that, for each person x, x loves y'. Sentence 6 is *not* true of the love-triangle just described. Again, suppose that our domain of discourse is restricted to Imre, Juan and Karl. Then this requires that there be someone whom Juan, Imre and Karl all 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.	$(\forall\exists)$
So:	There is some truth, that no person can know.	$(\exists\forall)$

This argument form is obviously invalid. It's just as bad as:

	Every dog has its day.	$(\forall\exists)$
So:	There is a day for all the dogs.	$(\exists\forall)$

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

21.4 Stepping-stones to symbolisation

Once we have the possibility of multiple quantifiers and many-place predicates, representation in FOL can quickly start to become a bit tricky. When you are trying to symbolise a complex sentence, it can be helpful to lay down several stepping stones. As usual, this idea is best illustrated by example. Consider this representation key:

domain: people and dogs
D: $\_\_\__1$ is a dog
F: $\_\_\__1$ is a friend of $\_\_\__2$
O: $\_\_\__1$ owns $\_\_\__2$
h: Hassan

And now let's try to symbolise these sentences:

7. Hassan is a dog owner.
8. Someone is a dog owner.
9. All of Hassan's friends are dog owners.
10. Every dog owner is a friend of a dog owner.
11. Every dog owner's friend owns a dog of a friend.

Sentence 7 can be paraphrased as, 'There is a dog that Hassan owns'. This can be symbolised by '$\exists x(Dx \wedge Ohx)$'.

Sentence 8 can be paraphrased as, 'There is some y such that y is a dog owner'. Dealing with part of this, we might write '$\exists y$(y is a dog owner)'. Now the fragment we have left as 'y is a dog owner' is much like sentence 7, except that it is not specifically about Hassan. So we can symbolise sentence 8 by:

$$\exists y\exists x(Dx \wedge Oyx)$$

We need to pause to clarify something here. In working out how to symbolise 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 symbolising the entire English sentence with a FOL sentence. You should regard it as a bit of rough-working-out, on a par with the doodles that you might absent-mindedly draw in the margin of this book, whilst you are concentrating fiercely on some problem.

Sentence 9 can be paraphrased as, 'Everyone who is a friend of Hassan is a dog owner'. Using our stepping-stone tactic, we might write

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

Now the fragment that we have left to deal with, 'x is a dog owner', is structurally just like sentence 7. But it would be a mistake for us simply to write:

$$\forall x\big[Fxh \to \exists x(Dx \wedge Oxx)\big]$$

This would leave us with 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 quantifier. But 'Dx' also falls under the scope of the existential quantifier '$\exists x$', so the 'x' in 'Dx' should be governed by that quantifier. And now confusion reigns: which quantifier is in charge of the 'x' in 'Dx'? There's no good answer to this question, and so logicians stipulate that their variables must never clash: no single variable should ever be governed by more than one quantifier at the same time.

To continue our symbolisation, then, we must choose some different variable for our existential quantifier. What we want is something like:

$$\forall x\big[Fxh \to \exists z(Dz \wedge Oxz)\big]$$

and this adequately symbolises sentence 9.

Sentence 10 can be paraphrased as 'For any x that is a dog owner, there is a dog owner who is a friend of x'. Using our stepping-stone tactic, this becomes

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

Completing the symbolisation, we end up with

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

Note that the variable 'z', is governed by one quantifier in the antecedent, and by another quantifier in the consequent. You might worry that this will let confusion creep back in, but actually, everything is OK: there is no clash of variables; each occurrence of 'z' is governed by just one quantifier. We might graphically represent the scope of the quantifiers like this:

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

This shows that no variable is being governed by two quantifiers at the same time.

Sentence 11 is the trickiest yet. First we paraphrase it as 'For any x that is a friend of a dog owner, x owns a dog which is also owned by a friend of x'. Using our stepping-stone tactic, this becomes:

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

We can now work on this a bit more:

$$\begin{aligned}&\forall x\big[\exists y(Fxy \wedge y \text{ is a dog owner}) \to\\ &\qquad\qquad \exists y(Dy \wedge Oxy \wedge y \text{ is owned by a friend of } x)\big]\end{aligned}$$

And a bit more:

$$\forall x\big[\exists y(Fxy \land \exists z(Dz \land Oyz)) \to \exists y(Dy \land Oxy \land \exists z(Fzx \land Ozy))\big]$$

And we are done!

Practice exercises

A. Using this symbolisation key:

domain: all animals
A: _____$_1$ is an alligator
M: _____$_1$ is a monkey
R: _____$_1$ is a reptile
Z: _____$_1$ lives at the zoo
L: _____$_1$ loves _____$_2$
a: Amos
b: Bouncer
c: Cleo

symbolise 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.
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 symbolisation key:

domain: all animals
C: _____$_1$ likes cartoons
D: _____$_1$ is a dog
L: _____$_1$ is larger than _____$_2$
b: Bertie
e: Emerson
f: Fergis

symbolise the following sentences in FOL:

1. Bertie is a dog who likes cartoons.
2. Bertie, Emerson, and Fergis are all dogs.
3. Emerson is larger than Bertie, and Fergis is larger than Emerson.
4. All dogs like cartoons.
5. Only dogs like cartoons.
6. There is a dog that is larger than Emerson.
7. If there is a dog larger than Fergis, then there is a dog larger than Emerson.

8. No animal that likes cartoons is larger than Emerson.
9. No dog is larger than Fergis.
10. Any animal that dislikes cartoons is larger than Bertie.
11. There is an animal that is between Bertie and Emerson in size.
12. There is no dog that is between Bertie and Emerson in size.
13. No dog is larger than itself.
14. 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 cartoons.

C. Using the following symbolisation key:

domain: people
D: $\underline{\quad\quad}_1$ dances ballet
F: $\underline{\quad\quad}_1$ is female
M: $\underline{\quad\quad}_1$ is male
C: $\underline{\quad\quad}_1$ is a child of $\underline{\quad\quad}_2$
S: $\underline{\quad\quad}_1$ is a sibling of $\underline{\quad\quad}_2$
a: Abebi
n: Naija
o: Orodena

symbolise the following sentences in FOL:

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

Identity 22

Consider this sentence:

1. Everyone owes money to Ebenezer

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

O: _____$_1$ owes money to _____$_2$
e: Ebenezer

we can symbolise sentence 1 by '$\forall x Oxe$'. But this has a somewhat odd consequence. It requires that *everyone* in the domain owes money to Ebenezer. But the domain must include Ebenezer himself. (Remeber that names can only ever refer to individuals in our domain.) So this entails that Ebenezer owes money to himself.

Perhaps we meant to say:

2. Everyone *other than Ebenezer* owes money to Ebenezer

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

22.1 Adding identity

The symbol '=' is a two-place predicate. Since it is to have a special meaning, we shall 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 symbolisation key:

=: _____$_1$ is identical to _____$_2$

(Since this is always part of our symbolisation keys, we don't actually need to bother explicitly including it in any of them.) Importantly, in this context, when we say that x is identical to y, we do not merely mean that they are indistinguishable, or that anything true of one of them is true of the other too. Rather, we mean that x is *the very same* object as y. (Philosophers and logicians call this NUMERICAL IDENTITY.)

Now suppose we want to symbolise this sentence:

3. Ebenezer is Mister Scrooge.

Let us add to our symbolisation key:

m: Mister Scrooge

Now sentence 3 can be symbolised as '$e = m$'. This means that the names 'e' and 'm' both refer to the same thing.

We can also now deal with sentence 2. This sentence can be paraphrased as 'Everyone who is not identical to Ebenezer owes money to Ebenezer'. Paraphrasing some more, we get: 'For all x, if x is not identical to Ebenezer, then x is owes money to Ebenezer'. Now that we are armed with our new identity symbol, we can symbolise this as '$\forall x(\neg x = e \rightarrow Oxe)$'.

This last sentence contains the formula '$\neg x = e$'. And 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 no problem. We are simply negating the entire formula, '$x = e$'. (In some textbooks, they write '$x \neq e$' rather than '$\neg x = e$'. That might be a little bit easier to look at, but it's worth sticking with '$\neg x = e$' for now; that way, we can see that we're dealing with a good old negation.)

Here are some more sentences whose symbolisation requires identity:

4. No one besides Ebenezer owes money to Bob.
5. Only Ebenezer owes Bob money.

Letting 'b' name Bob, sentence 4 can be paraphrased as, 'No one who is not Ebenezer owes money to Bob'. This can be symbolised by '$\neg\exists x(\neg x = e \wedge Oxb)$'. Equally, sentence 4 can be paraphrased as 'For all x, if x owes money to Bob, then x is Ebenezer'. Then it can be symbolised as '$\forall x(Oxb \rightarrow x = e)$'.

Sentence 5 can be treated similarly to sentence 4. But there is a subtlety here. Does either sentence entail that Ebenezer himself owes money to Bob? It isn't entirely obvious. (My own linguistic intuition is that sentence 4 doesn't entail that Ebenezer owes Bob money, and sentence 5 does. But you might reasonably disagree!) What is certain, though, is that '$\forall x(Oxe \rightarrow x = b)$' *doesn't* entail it: this FOL sentence only tells us that *if* someone owes money to Bob, then that someone is Ebenezer. But no matter! If we do want to imply that Ebenezer does in fact owe money to Bob, then we can just add that as a conjunct, like this: '$Oeb \wedge \forall x(Oxb \rightarrow x = e)$'.

22.2 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:

6. There is at least one apple
7. There are at least two apples
8. There are at least three apples

We shall use the symbolisation key:

A: $\underline{\quad\quad}_1$ is an apple

Sentence 6 does not require identity. It can be adequately symbolised by '$\exists x Ax$': there is some apple; perhaps many, but at least one.

It might be initially tempting to symbolise sentence 7 without identity as '$\exists x\exists y(Ax \wedge Ay)$'. But that wouldn't be quite right. Roughly, this FOL sentence 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 to use the identity predicate. Sentence 7 needs to say that the two apples that exist are not identical, so it can be symbolised by '$\exists x \exists y(Ax \land Ay \land \neg x = y)$'.

Sentence 8 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: '$\exists x \exists y \exists z(Ax \land Ay \land Az \land \neg x = y \land \neg y = z \land \neg x = z)$'.

22.3 There are at most...

Now consider these sentences:

9. There is at most one apple
10. There are at most two apples

Sentence 9 can be paraphrased as, 'It is not the case that there are at least *two* apples'. This is just the negation of sentence 7:

$$\neg \exists x \exists y(Ax \land Ay \land \neg x = y)$$

But sentence 9 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 symbolised by

$$\forall x \forall y \big[(Ax \land Ay) \to x = y\big]$$

The two sentences will turn out to be logically equivalent.

In a similar way, sentence 10 can be approached in two equivalent ways. It can be paraphrased as, 'It is not the case that there are at least *three* apples', so we can offer:

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

Or, 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:

$$\forall x \forall y \forall z \big[(Ax \land Ay \land Az) \to (x = y \lor x = z \lor y = z)\big]$$

22.4 There are exactly...

We can now consider precise statements, like:

11. There is exactly one apple.
12. There are exactly two apples.
13. There are exactly three apples.

Sentence 11 can be paraphrased as, 'There is *at least* one apple and there is *at most* one apple'. This is just the conjunction of sentence 6 and sentence 9. So we can offer:

$$\exists x Ax \land \forall x \forall y \big[(Ax \land Ay) \to x = y\big]$$

But it is perhaps more straightforward to paraphrase sentence 11 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:

$$\exists x \big[Ax \land \forall y(Ay \to x = y)\big]$$

Similarly, sentence 12 may be paraphrased as, 'There are *at least* two apples, and there are *at most* two apples'. Thus we could offer

$$\exists x \exists y(Ax \land Ay \land \neg x = y) \land \forall x \forall y \forall z \big[(Ax \land Ay \land Az) \to (x = y \lor x = z \lor y = z)\big]$$

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:

$$\exists x \exists y \big[Ax \land Ay \land \neg x = y \land \forall z(Az \to (x = z \lor y = z)\big]$$

Finally, consider these sentence:

14. There are exactly two things
15. There are exactly two objects

It might be tempting to add a predicate to our symbolisation key, to symbolise 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 (i.e. every *thing*). So we can symbolise them both like this:

$$\exists x \exists y \neg x = y \land \neg \exists x \exists y \exists z(\neg x = y \land \neg y = z \land \neg x = z)$$

Or, equivalently but a bit more snappily, like this:

$$\exists x \exists y \big[\neg x = y \land \forall z(x = z \lor y = z)\big]$$

Practice exercises

A. Explain why:

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

Definite descriptions 23

Consider sentences like:

1. The keyboardist was born in Oxford.
2. Jonny is the keyboardist.
3. The keyboardist is the composer.

These sentences use definite descriptions. Definite descriptions are expressions like 'the keyboardist' and 'the composer', which are meant to describe a *unique* object. (They should be contrasted with *indefinite* descriptions, like 'a keyboardist', which do not imply uniqueness.) The question we face is: how should we deal with definite descriptions in FOL?

One option would be to introduce new names whenever we come across a definite description. For example, we could try symbolising 'the keyboardist' with the FOL name 'k'. However, there is a problem with this simple idea. We know that '*the* keyboardist' should stand for *a* keyboardist (if it stands for anyone at all). We want to preserve that information in our symbolisation. But the name 'k' doesn't encode any information about the object it stands for.

So how should we symbolise definite descriptions in FOL? There are a few different workable options, but we'll focus on the one suggested by Bertrand Russell in his famous paper, 'On denoting'.[1] What makes Russell's suggestion so appealing is that it won't require us to add any new basic machinery to FOL.

23.1 Russell's analysis

Russell's analysis of definite descriptions is based on the following insight: when we say 'the F is G', our aim is to pick out the *one and only* thing that is F, and then say of it that it is G. Thus Russell analysed the notion of a definite description as follows:

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.

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

Now, one might worry that I can say 'the keyboardist was born in Oxford' without implying that there is one and only one keyboardist in the universe. What I mean is just that the keyboardist *in the band we are talking about* was born in Oxford. But this is not really a counterexample to Russell's analysis. It just reminds us that we need to check what the domain of discourse is. If I say 'the keyboardist was born in Oxford', the domain of discourse is likely to be restricted by context to one particular band (or maybe to the people in the room, or to the people I know, or...).

If we accept Russell's analysis of definite descriptions, then we can symbolise 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:

$$\exists x Fx \land \forall x \forall y((Fx \land Fy) \to x = y) \land \forall x(Fx \to Gx)$$

In fact, we could express the same point rather more crisply, by recognising 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:

$$\exists x\big[Fx \land \forall y(Fy \to x = y) \land Gx\big]$$

Using these sorts of techniques, we can now symbolise sentences 1–3:

1. The keyboardist was born in Oxford.
2. Jonny is the keyboardist.
3. The keyboardist is the composer.

Let's start, as always, with a symbolisation key:

domain: members of Radiohead
K: $\_\_\_\_\__1$ is a keyboardist
O: $\_\_\_\_\__1$ was born in Oxford
C: $\_\_\_\_\__1$ is a composer
j: Jonny Greenwood

Sentence 1 is exactly like the examples we have just considered. So we would symbolise it by '$\exists x(Kx \land \forall y(Ky \to x = y) \land Ox)$'.

Sentence 2 poses no problems either: '$\exists x(Kx \land \forall y(Ky \to x = y) \land x = j)$'.

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 keyboardist, x, and there is exactly one composer, y, and x = y'. So we can symbolise it by:

$$\exists x \exists y\big(\big[Kx \land \forall z(Kz \to x = z)\big] \land \big[Cy \land \forall z(Cz \to y = z)\big] \land x = y\big)$$

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

23.2 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 to stand for the present King of France, then everything would go wrong: remember from §19 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 §20), 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 John Blechl, who runs *Beginning Philosophy*, claimed:

4. I went for dinner with the present King of France

Using the following symbolisation key:

K: $\_\_\_\_\__1$ is a present King of France
D: $\_\_\_\_\__1$ went for dinner with $\_\_\_\_\__2$
a: John Blechl

Sentence 4 would be symbolised 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 John went for dinner with him.
6. There is a unique present King of France, but John did not go for dinner with him.

Sentence 5 might be paraphrased by 'It is not the case that: John went for dinner with the present King of France'. It will then be symbolised 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 symbolised 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 did not go to dinner with John.

23.3 Is Russell's analysis adequate?

How good is Russell's analysis of definite descriptions? This question has generated a substantial philosophical literature.[3] Unfortunately, though, it would take us too far off topic to get into that literature here. (If you're interested, why not take the *Philosophy of Language* module next year?)

But here's what we can say now. Russell's analysis allows us to handle lots of the ways that we use definite descriptions, without needing to add any fancy new devices to FOL. That's good enough for us!

[2]Stephen Neale, *Descriptions*, 1990, Cambridge, MA: MIT Press.

[3]Here are three classics: P.F. Strawson, 'On Referring', 1950, *Mind* 59, pp. 320–34; Keith Donnellan, 'Reference and Definite Descriptions', 1966, *Philosophical Review* 77, pp. 281–304; 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.

Practice exercises

A. Using the following symbolisation key:

domain: people
K: _____$_1$ knows the combination to the safe
S: _____$_1$ is a spy
V: _____$_1$ is a vegetarian
T: _____$_1$ trusts _____$_2$
h: Hofthor
i: Ingmar

symbolise 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 symbolisation key:

domain: cards in a standard deck
B: _____$_1$ is black
C: _____$_1$ is a club
D: _____$_1$ is a deuce
J: _____$_1$ is a jack
M: _____$_1$ is a man with an axe
O: _____$_1$ is one-eyed
W: _____$_1$ is wild

symbolise 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 symbolisation key:

domain: animals in the world
B: _____$_1$ is in Farmer Brown's field
H: _____$_1$ is a horse
C: _____$_1$ is a cow
F: _____$_1$ is faster than _____$_2$
r: Redrum

symbolise 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. Every horse is faster than every cow.
5. Redrum is faster than every cow in Farmer Brown's field.
6. There is a cow in Farmer Brown's field that is faster than a horse in Farmer Brown's field.
7. Redrum is faster than every other horse.
8. The fastest horse is in Farmer Brown's field.
9. The fastest horse in Farmer Brown's field is faster than Redrum.
10. The fastest horse in Farmer Brown's field is faster than the fastest cow in the world.

D. In this section, we symbolised 'Jonny is the keyboardist' by '$\exists x(Kx \land \forall y(Ky \to x = y) \land x = j)$'. Two equally good symbolisations would be:

- $Kj \land \forall y(Ky \to y = j)$
- $\forall y(Ky \leftrightarrow y = j)$

Explain why these would be equally good symbolisations.

Sentences of FOL 24

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

24.1 Expressions

There are six kinds of symbols in FOL:

Predicates (with superscripts)	$=, A^1, B^1, \ldots, Z^1, A^2, B^2, \ldots$
and with subscripts, as needed	$A^1_1, B^2_1, Z^5_1, A^8_2, J^{10}_{25}, \ldots$
Names	$a, b, c, \ldots, r$
with subscripts, as needed	$a_1, b_{224}, h_7, m_{32}, \ldots$
Variables	s, t, u, v, w, x, y, z
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.

24.2 Terms and formulas

In §6, we went straight from the statement of the vocabulary of TFL to the definition of a sentence of TFL. In FOL, we shall have to go via an intermediary stage: via the notion of a *formula*. The intuitive idea is that a formula is any sentence, or anything which can be turned into a sentence by adding quantifiers out front. But this will take some unpacking.

We start by defining the notion of a TERM.

> A TERM is any name or any variable.

So, here are some terms:

$$a, b, x, x_1 x_2, y, y_{254}, z$$

We next need to define ATOMIC FORMULAS.

1. If $\mathcal{R}^n$ is an n-place predicate and $\mathcal{t}_1, \mathcal{t}_2, \ldots, \mathcal{t}_n$ are terms, then $\mathcal{R}^n \mathcal{t}_1 \mathcal{t}_2 \ldots \mathcal{t}_n$ is an atomic formula.
2. If $\mathcal{t}_1$ and $\mathcal{t}_2$ are terms, then $\mathcal{t}_1 = \mathcal{t}_2$ is an atomic formula.
3. Nothing else is an atomic formula.

The use of swashfonts here follows the conventions laid down in §7. So, '$\mathcal{R}^n$' is not itself a predicate of FOL. Rather, it is a symbol of our metalanguage (augmented English) that we use to talk about any predicate of FOL. Similarly, '$\mathcal{t}_1$' is not a term of FOL, but a symbol of the metalanguage that we can use to talk about any term of FOL. Here are some atomic formulas:

$$x = a$$
$$a = b$$
$$F^1 x$$
$$F^1 a$$
$$G^3 xay$$
$$G^3 aaa$$
$$S^6 x_1 x_2 abyx_1$$
$$S^6 by_{254} zaaz$$

(We have re-attached the superscripts to indicate how many argument places each predicate has. We really do need these superscripts in order to apply our formal definition of 'atomic formula'. But don't worry, we'll stop using them as soon as we can get away with it!)

Once we know what atomic formulas are, we can offer recursion clauses to define arbitrary formulas. The first few clauses are exactly the same as for TFL.

1. Every atomic formula is a formula.
2. If $\mathcal{A}$ is a formula, then $\neg\mathcal{A}$ is a formula.
3. If $\mathcal{A}$ and $\mathcal{B}$ are formulas, then $(\mathcal{A} \wedge \mathcal{B})$ is a formula.
4. If $\mathcal{A}$ and $\mathcal{B}$ are formulas, then $(\mathcal{A} \vee \mathcal{B})$ is a formula.
5. If $\mathcal{A}$ and $\mathcal{B}$ are formulas, then $(\mathcal{A} \rightarrow \mathcal{B})$ is a formula.
6. If $\mathcal{A}$ and $\mathcal{B}$ are formulas, then $(\mathcal{A} \leftrightarrow \mathcal{B})$ is a formula.
7. If $\mathcal{A}$ is a formula, χ is a variable, $\mathcal{A}$ contains at least one occurrence of χ, and $\mathcal{A}$ contains neither $\forall\chi$ nor $\exists\chi$, then $\forall\chi\mathcal{A}$ is a formula.
8. If $\mathcal{A}$ is a formula, χ is a variable, $\mathcal{A}$ contains at least one occurrence of χ, and $\mathcal{A}$ contains neither $\forall\chi$ nor $\exists\chi$, then $\exists\chi\mathcal{A}$ is a formula.
9. Nothing else is a formula.

So here are some more formulas:

$$F^1x$$
$$G^3ayz$$
$$S^6yzyayx$$
$$(G^3ayz \rightarrow S^6yzyayx)$$
$$\forall z(G^3ayz \rightarrow S^6yzyayx)$$
$$F^1x \leftrightarrow \forall z(G^3ayz \rightarrow S^6yzyayx)$$
$$\exists y(F^1x \leftrightarrow \forall z(G^3ayz \rightarrow S^6yzyayx))$$
$$\forall x\exists y(F^1x \leftrightarrow \forall z(G^3ayz \rightarrow S^6yzyayx))$$

But this is *not* a formula:

$$\forall x\exists xG^3xxx$$

Certainly 'G^3xxx' is a formula. And '$\exists xG^3xxx$' is therefore also a formula. But we cannot form a new formula by putting '$\forall x$' at the front. This violates the constraints on clause 7 of our recursive definition: '$\exists xG^3xxx$' contains at least one occurrence of 'x', but it already contains '$\exists x$'.

These constraints have the effect of ensuring that variables can only be governed by one quantifier at a time. And in fact, we can now give a formal definition of scope, which incorporates the definition of the scope of a quantifier. First, recall the definition of the 'main logical operator' that we gave way back in §6.2:

The MAIN LOGICAL OPERATOR in a formula is the operator that was introduced last, when that formula was constructed using the recursion rules.

When we first gave this definition, we were working in TFL, and so the only operators we had in mind were sentential connectives. Now we need to include quantifiers. So *both* quantifiers and connectives count as operators.

Earlier, I gave a method for finding which operator is the main operator of a formula, but that method didn't take quantifiers into account. Here is the new method, which does. To use this method, you have to make sure that you have included *all* brackets, even ones you can usually get away with deleting (see §24.4).

- **Step 1.** Check if the first symbol in the formula is '$\neg$'; if so, then that '$\neg$' is the main logical operator.
- **Step 2.** If not, check if the first symbol in the formula is a quantifier; if so, then that quantifier is the main logical operator.
- **Step 3.** If not, then start counting brackets. Open-brackets '(' are worth $+1$, close-brackets ')' are worth -1. The first connective you hit which isn't a '$\neg$' or a quantifier when your count is at exactly 1 is the main logical operator.

Here are three examples:

1. $\neg\exists x(Fx \wedge Gy)$
2. $\exists x\neg(Fx \wedge Gy)$
3. $(\exists x\neg(Fx \wedge Gy) \rightarrow \forall z\neg Fz)$

The first symbol in formula 1 is '$\neg$', so that is the main operator. The first symbol in formula 2 is the quantifier '$\exists x$', so that is the main operator. Formula 3 doesn't start with a quantifier or a '$\neg$', so we need to start counting brackets:

$$(^1\exists x\neg(^2Fx \wedge Gx)^1 \rightarrow \forall z\neg Fz)^0$$

The first symbol we hit when the count is at 1 is '$\exists x$', but the rule is that we ignore quantifiers when we're counting brackets. The next symbol we hit is '$\neg$', but we also ignore negations when we're counting. The next operator we hit when the count is back at 1 is '$\rightarrow$', so that is the main operator.

Now we can define what we mean by the 'scope' of an operator:

> The SCOPE of a logical operator in a formula is the subformula for which that operator is the main logical operator.

We can graphically illustrate the scope of the quantifiers in '$\forall x\exists y(Fx \leftrightarrow \forall z(Gayz \rightarrow Syzyayx))$' as follows:

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

24.3 Sentences

Recall that, in logic, we focus on *assertoric* sentences: sentences that can be either true or false. Many formulas are not sentences. Consider the following symbolisation key:

domain: people
L: _____$_1$ loves _____$_2$
d: Danny

Consider the atomic formula 'Lzz'. All atomic formula are formulas, so 'Lzz' is a formula. But can it be true or false? You might think that it will be true just in case the person named by 'z' loves themself, in the same way that 'Ldd' is true just in case Danny (the person named by 'd') loves himself. *But 'z' is a variable, and does not name anyone or anything.*

Of course, if we put an existential quantifier out front, obtaining '$\exists z Lzz$', then this would be true iff someone loves themself. Equally, if we wrote '$\forall z Lzz$', this would be true iff everyone loves themself. The point is that we need a quantifier to tell us how to deal with a variable.

Let's make this idea precise.

> A BOUND VARIABLE is an occurrence of a variable χ that is within the scope of either $\forall\chi$ or $\exists\chi$.
>
> A FREE VARIABLE is any variable that is not bound.

For example, consider the formula

$$\forall x(Ex \lor Dy) \to \exists z(Ex \to Lzx)$$

The scope of the universal quantifier '$\forall x$' is '$\forall x(Ex \lor Dy)$', so the first 'x' is bound by the universal quantifier. However, the second and third occurrences of 'x' are free. Equally, the 'y' is free. The scope of the existential quantifier '$\exists z$' is '$(Ex \to Lzx)$', so 'z' is bound.

Finally we can say the following:

> A SENTENCE of FOL is any formula of FOL that contains no free variables.

24.4 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.[1] For example, this

[1] If you are trying to identify the main operator in a sentence without outermost brackets, you need to tweak the procedure we gave in §24.2. First, you should skip Steps 1 and 2, because if the outermost brackets have been deleted, then negations and quantifiers can't be the main operator. So, jump straight to Step 3, but now you are looking for the first operator (other than negations and quantifiers) you reach when the bracket-count is 0.

What should you do if you're trying to identify the main connective, but you aren't sure whether the outermost brackets have been included or not? First, try assuming that the outermost brackets have *not* been included. If you find an operator which isn't a negation or a quantifier when the bracket-count is 0, then that's the main connective. But if you don't, then that means that the formula does include its outermost brackets, and so you should go back to the original three-step method.

formula,

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

can be written as:

$$\exists x Fx \leftrightarrow \forall y Gy$$

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

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

can be written as:

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

Third, we may omit brackets between each pair of conjuncts when writing long series of conjunctions. For example, this formula,

$$\exists v \exists w \exists x \exists y \exists z(Fv \land (Fw \land (Fx \land (Fy \land Fz))))$$

can be written as:

$$\exists v \exists w \exists x \exists y \exists z(Fv \land Fw \land Fx \land Fy \land Fz)$$

Fourth, we may omit brackets between each pair of disjuncts when writing long series of disjunctions. For example, this formula,

$$\exists v \exists w \exists x \exists y \forall z(z = v \lor (z = w \lor (z = x \lor z = y)))$$

can be written as:

$$\exists v \exists w \exists x \exists y \forall z(z = v \lor z = w \lor z = x \lor z = y)$$

Practice exercises

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

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

Chapter 6

Interpretations

Extensionality 25

The aim of this section is to introduce the idea of an *interpretation* of FOL. Then, in the rest of this chapter, we'll put that idea to good work.

25.1 Symbolising versus translating

Back when we introduced TFL, we emphasised that it 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 symbolisation key, e.g.:

P: Big Ben is in London

But 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 emphasised is that TFL cannot handle differences in meaning that go beyond mere differences in truth-value.

FOL has some similar limitations. It gets beyond mere truth-values, since it allows us to split up sentences into terms, predicates and quantifier expressions. This enables us to consider what is *true of* a particular object, or of some objects, or of all objects. But we can do no more than that.

When we provide a symbolisation key for some FOL predicates, such as:

Y: $\_\_\_\_\__1$ leads a First-Year Philosophy module at the University of York in Autumn 2022

we do not carry the *meaning* of the English predicate across into our FOL predicate. We are simply stipulating something like the following:

- 'Y' and '$\_\_\_\_\__1$ leads a First-Year Philosophy module at the University of York in Autumn 2022' are to be *true of* exactly the same things (whatever they may be)

So, in particular:

- 'Y' is to be true of all and only the leaders of First-Year Philosophy modules at the University of York in Autumn 2022.

This is an *indirect* stipulation. Alternatively we can stipulate predicate extensions *directly*. We can stipulate that 'Y' is to be true of, and only of, Rob Trueman and John Blechl. As it happens, this direct stipulation would have the same effect as the indirect stipulation. But note that the English predicates '_____ is either Rob Trueman or John Blechl' and '_____ leads a First-Year Philosophy module at the University of York in Autumn 2022' 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. This is normally summed up by saying that FOL is an EXTENSIONAL LANGUAGE.

For this reason, we will say only that FOL sentences *symbolise* English sentences. It is doubtful that we are *translating* English into FOL, for translations should preserve meanings.

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 'H' should be true of, and only of, the following objects:

Danny DeVito
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 'H', suppose we now add these entries to our symbolisation key:

d: Danny DeVito
m: Michael Keaton
p: the number π

Then 'Hd' and 'Hp' will both be true, on this interpretation, but 'Hm' will be false, since Michael Keaton was not among the stipulated objects.

This process of explicit stipulation is sometimes described as stipulating the *extension* of a predicate.

> The EXTENSION of a predicate is the collection of things which that predicate is true of.

25.3 Many-place predicates

All of this is quite easy to understand when it comes to one-place predicates. But it gets much messier when we consider two-place predicates. Consider a symbolisation key like:

L: $\_\_\_\_\__1$ loves $\_\_\_\_\__2$

Given what we said above, this symbolisation key should be read as saying:

- ‘L’ and ‘$\underline{\hspace{2em}}_1$ loves $\underline{\hspace{2em}}_2$’ are to be true of exactly the same things

So, in particular:

- ‘L’ 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 is not always requited (sadly). So if Daniel loves Simon but not *vice versa*, then ‘L’ is true of Daniel and Simon in that order, but it is not true of Simon and Daniel in *this* order.

That is an indirect stipulation. What about a direct stipulation? This is slightly harder. If we *simply* list objects that fall under ‘L’, 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 ‘B’ is to be true of, and only of, the following pairs of objects:

<Lenny Bruce, Richard Pryor>
<Daniel Kitson, John Kearns>
<John Kearns, Daniel Kitson>
<Richard Pryor, Richard Pryor>

Here the angle-brackets keep us informed concerning order. Suppose we now add the following stipulations:

d: Daniel Kitson
j: John Kearns
l: Lenny Bruce
r: Richard Pyror

Then ‘Blr’ will be true, since <Lenny Bruce, Richard Pryor> was on our explicit list. But ‘Brl’ will be false, since <Richard Pryor, Lenny Bruce> was not on our list. However, both ‘Bdj’ and ‘Bjd’ will be true, since both <Daniel Kitson, John Kearns> and <John Kearns, Daniel Kitson> are on our explicit list. Finally, ‘Brr’ will be true, since <Richard Pryor, Richard Pryor> is on our explicit list, but ‘Bdd’, ‘Bjj’ and ‘Bll’ will all be false.

We do something similar for predicates with even more argument places. Suppose that ‘C’ is a *three-place* predicate. In that case, we would directly stipulate its extension by listing a bunch of ordered *triples*, like this:

<Lenny Bruce, Richard Pryor, Lenny Bruce>
<John Kearns, Lenny Bruce, Richard Pryor>
<Daniel Kitson, Richard Pyror, Lenny Bruce>

Now ‘$Clrl$’, ‘$Cjlr$’ and ‘$Cdrl$’ will all be true, but every other sentence you can build by combining ‘C’ with ‘d’, ‘j’, ‘l’ or ‘r’ will be false.

To make these ideas more precise, we would need to develop some *set theory*. This would give you some apparatus for dealing with extensions and with ordered pairs (and ordered triples, etc.) However, set theory is not covered in this book. So, for now, we will talk in a relatively informal way about extensions and the like. The general idea should hopefully be clear enough.

25.4 Interpretations

We defined a *valuation* in TFL as any assignment of truth and falsity to atomic sentences. We then presented a series of semantic clauses which determined the truth-values of complex sentences on each valuation. Now we will do something similar for FOL. However, in FOL we will work with INTERPRETATIONS, rather than valuations.

An interpretation consists 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

(We don't need to bother specifying which objects '=' is true of, since it *always* symbolises '$\underline{\hspace{2em}}_1$ is identical to $\underline{\hspace{2em}}_2$'.)

Clearly, the symbolisation keys that we considered in chapter 5 give us one very convenient way to present an interpretation. But I do not want that to give the wrong impression. Our interpretations do not need to give us a neat way of reading FOL sentences in English. If we want to, we can specify interpretations completely artificially: all we need to do is specify which objects are in the domain, and which objects are in the extension of each predicate. So as well as presenting interpretations like this,

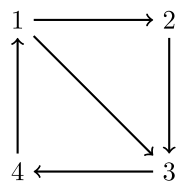
domain: cities in France
S: $\underline{\hspace{2em}}_1$ is in southern France
C: $\underline{\hspace{2em}}_1$ is the capital of France
F: $\underline{\hspace{2em}}_1$ was founded in 2017
L: $\underline{\hspace{2em}}_1$ is larger than $\underline{\hspace{2em}}_2$

we can also present interpretations like this:

domain: Paris, Toulouse, Marseille
S: Toulouse, Marseille
C: Paris
F^1:
L: <Paris, Toulouse>, <Paris, Marseille>, <Marseille, Toulouse>

This interpretation should be understood as directly stipulating which objects are to be in our domain, and which objects are to be in the extensions of the listed predicates. So for example, Toulouse and Marseille are both in the extension of 'S', but nothing else is. What about 'F'? I left the entry for 'F' empty, and I meant that to indicate that 'F' has *nothing at all* in its extension; 'F' is not true of anything in the domain. (I also included the superscript on 'F', because otherwise you couldn't tell that it was an empty one-place predicate, rather than, e.g., an empty two-place predicate.) The final entry is for a two-place predicate 'L', which is why we had to use angle brackets: 'L' is true of Paris and Toulouse in that order, but it is not true of Toulouse and Paris in *this* order.

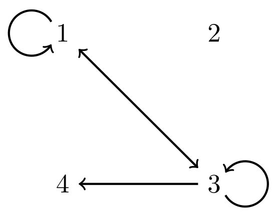
We can also present interpretations *diagrammatically*. Suppose we want to consider just a single two-place predicate, 'R'. Then we can represent it just by drawing an arrow between two objects, and stipulate that '*R*' is to hold between x and y (in that order) just in case there is an arrow running from x to y in our diagram. As an example, we might offer:



Drawing this diagram is another way of presenting this interpretation:

domain: 1, 2, 3, 4
R: <1,2>, <2,3>, <3,4>, <4,1>, <1,3>

Here is another example, to help you get the idea:



Drawing this diagram is a way of presenting the following interpretation:

domain: 1, 2, 3, 4
R: <1,3>, <3,1>, <3,4>, <1,1>, <3,3>

If we wanted, we could make our diagrams more complex. For example, we could add names as labels for particular objects. Equally, to symbolise 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 that predicate.

Truth in FOL 26

We know what interpretations are. It's now time to put them to work. In particular, in this section we will use interpretations to provide a detailed account of what it is for an arbitrary sentence of FOL to be true or false.

As we explained in §24, 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 work up to a completely general explanation in this section. However, to try to keep the explanation comprehensible, we will often use the following interpretation:

domain: all people born before 2000CE
G: _____$_1$ is Greek
B: _____$_1$ was born before _____$_2$
p: Plato
r: Bertrand Russell

This will be our *Example Interpretation* in what follows.

26.1 Atomic sentences

The truth of atomic sentences should be fairly straightforward. The sentence 'Gp' should be true just in case 'G' is true of the object named by 'p'. Given our Example Interpretation, this is true iff Plato is Greek. Plato is Greek. So the sentence is true. Equally, 'Gr' is false on our Example Interpretation.

Likewise, on this interpretation, 'Bpr' is true iff the object named by 'p' was born before the object named by 'r'. Well, Plato was born before Russell. So 'Bpr' is true. Equally, 'Bpp' is false: Plato was not born before Plato.

Dealing with atomic sentences, then, is very intuitive. When $\mathcal{R}^n$ is an n-place predicate and $a_1, a_2, \ldots, a_n$ are names:

$\mathcal{R}^n a_1 a_2 \ldots a_n$ is true in an interpretation **iff** $\mathcal{R}^n$ is true of the objects named by $a_1, a_2, \ldots, a_n$ in that interpretation (considered in that order)

Recall, though, that there is a second kind of atomic sentence: two names connected by an identity sign constitute an atomic sentence. This kind of atomic sentence is also easy to handle. 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 Example Interpretation, '$p = r$' is false, since Plato is distinct from Russell, but '$p = p$' and '$r = r$' are both true.

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:

> $\mathcal{A} \wedge \mathcal{B}$ is true in an interpretation **iff**
> both $\mathcal{A}$ is true and $\mathcal{B}$ is true in that interpretation;
> otherwise, $\mathcal{A} \wedge \mathcal{B}$ is false in that interpretation.
>
> $\mathcal{A} \vee \mathcal{B}$ is true in an interpretation **iff**
> either $\mathcal{A}$ is true or $\mathcal{B}$ is true (or both) in that interpretation;
> otherwise, $\mathcal{A} \vee \mathcal{B}$ is false in that interpretation.
>
> $\neg\mathcal{A}$ is true in an interpretation **iff**
> $\mathcal{A}$ is false in that interpretation;
> otherwise, $\neg\mathcal{A}$ is false in that interpretation.
>
> $\mathcal{A} \rightarrow \mathcal{B}$ is true in an interpretation **iff**
> either $\mathcal{A}$ is false or $\mathcal{B}$ is true (or both) in that interpretation;
> otherwise, $\mathcal{A} \rightarrow \mathcal{B}$ is false in that interpretation.
>
> $\mathcal{A} \leftrightarrow \mathcal{B}$ is true in an interpretation **iff**
> $\mathcal{A}$ has the same truth value as $\mathcal{B}$ in that interpretation;
> otherwise, $\mathcal{A} \leftrightarrow \mathcal{B}$ is false in that interpretation.

This presents the very same information as the characteristic truth tables for the connectives; it just does it in a slightly different way. Some examples will probably help to illustrate the idea. On our Example Interpretation:

- '$p = p \wedge Gp$' is true
- '$Bpr \wedge Gr$' is false because, although 'Bpr' is true, 'Gr' is false
- '$p = r \vee Gp$' is true
- '$\neg p = r$' is true
- '$Gp \wedge \neg(p = r \wedge Bpr)$' is true, because '$Gp$' is true and '$p = r$' is false

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*. And in fact, expressing the truth-conditions for quantified sentences is a bit more fiddly than one might expect.

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

Unfortunately, this naïve first thought is not general enough. For example, we want to be able to say that '$\forall x \exists y Bxy$' is true just in case the predicate '$\exists y B\_y$' is true of everything in the domain. And this is problematic, since our interpretation does not directly specify what '$\exists y B\_y$' is to be true of. Instead, whether or not this is true of something should follow just from the interpretation of 'B', the domain, and the meanings of the quantifiers.

So here is a naïve second thought. We might try to say that '$\forall x \exists y Bxy$' is to be true in an interpretation iff '$\exists y B\mathscr{a}y$' is true for *every* name $\mathscr{a}$ that we have included in our interpretation. And similarly, we might try to say that '$\exists y B\mathscr{a}y$' is true just in case '$B\mathscr{a}\mathscr{b}$' 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 Example Interpretation, we have only given interpretations for *two* names, 'p' and 'r'. But the domain — all people born before the year 2000CE — contains many more than two people. (Let's not try 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. I shall offer a few examples of how this might work, centring on our Example Interpretation, and I shall then present the formal definition.

In our Example Interpretation, '$\exists x Brx$' should be true. After all, in the domain, there is certainly someone who was born after Bertrand Russell. Lady Gaga is one of those people. Indeed, if we were to extend our Example Interpretation—temporarily, mind—by adding the name 'c' to refer to Lady Gaga, then 'Brc' would be true on this extended interpretation. And this, surely, should suffice to make '$\exists x Brx$' true on the original Example Interpretation.

In our Example Interpretation, '$\exists x(Gx \wedge Bxp)$' should also be true. After all, in the domain, there is certainly someone who was both Greek and born before Plato. Socrates is one such person. Indeed, if we were to extend our Example Interpretation by letting a new name, 'c', denote Socrates, then '$Gc \wedge Bcp$' would be true on this extended interpretation. Again, this should surely suffice to make '$\exists x(Gx \wedge Bxp)$' true on the original Example Interpretation.

In our Example Interpretation, '$\forall x \exists y Bxy$' should be false. After all, consider the last person born in the year 1999. I don't know who that was, but if we were to extend our Example Interpretation by letting a new name, 'c', denote that person, then we would not be able to find anyone else in the domain to denote with some further new name, perhaps 'd', in such a way that

'Bcd' would be true. Indeed, no matter *whom* we named with 'd', 'Bcd' would be false. And this observation is surely sufficient to make '$\exists yBcy$' *false* in our extended interpretation. And this, in turn, is enough to make '$\forall x \exists yBxy$' false on the original Example Interpretation.

If you have understood these three examples, then 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 $\mathcal{A}$ is a formula containing at least one instance of the variable χ, and that χ is free in $\mathcal{A}$. We will write this thus:

$$\mathcal{A}(\ldots \chi \ldots \chi \ldots)$$

Suppose also that c is a name. Then we shall write:

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

for the formula obtained by replacing *every* occurrence of χ in $\mathcal{A}$ with c. The resulting formula is called a SUBSTITUTION INSTANCE of $\forall \chi \mathcal{A}$ and $\exists \chi \mathcal{A}$. 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. Let c be a new name we add to our language. The sentence $\forall \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots)$ will be true iff $\mathcal{A}(\ldots c \ldots c \ldots)$ is true no matter what object (in the domain) we name with c. Similarly, the sentence $\exists \chi \mathcal{A}$ will be true iff there is *some* way to assign the name c to an object that makes $\mathcal{A}(\ldots c \ldots c \ldots)$ true. More precisely, we stipulate:

Let c be a new name added to the language.

$\forall \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots)$ is true in an interpretation **iff** $\mathcal{A}(\ldots c \ldots c \ldots)$ is true in *every* interpretation that extends the original interpretation by assigning an object to c (without changing the interpretation in any other way);
otherwise, $\forall \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots)$ is false in that interpretation.

$\exists \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots)$ is true in an interpretation **iff** $\mathcal{A}(\ldots c \ldots c \ldots)$ is true in *some* interpretation that extends the original interpretation by assigning an object to c (without changing the interpretation in any other way);
otherwise, $\exists \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots)$ is false in that interpretation.

To be clear: all this is doing is formalising (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:

domain: Corwin, Benedict
A: Corwin, Benedict
B: Benedict
N^1:
c: 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. $\exists x Bx \rightarrow \forall x Ax$

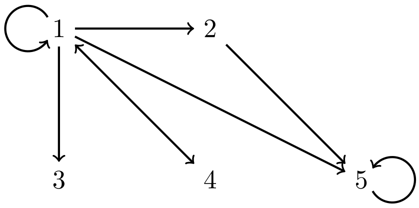
B. Consider the following interpretation:

domain: Luda, Capriana, Edgar
G: Luda, Capriana, Edgar
H: Capriana
M: Luda, Edgar
c: Capriana
e: Edgar

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. $\forall x \forall y \forall z((Rxy \wedge Ryz) \to Rxz)$
6. $\forall x \forall y \forall z((Rxy \wedge Rxz) \to Ryz)$
7. $\exists x \forall y \neg Rxy$
8. $\forall x(\exists y Rxy \to \exists y Ryx)$
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:

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

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

$$\vDash \mathcal{A}$$

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

An FOL sentence $\mathcal{A}$ is a TRUTH OF FOL iff $\mathcal{A}$ is true in every interpretation; i.e., $\vDash \mathcal{A}$.

$\mathcal{A}$ is a CONTRADICTION IN FOL iff $\mathcal{A}$ is false in every interpretation; i.e., $\vDash \neg\mathcal{A}$.

Two FOL sentences $\mathcal{A}$ and $\mathcal{B}$ are EQUIVALENT IN FOL iff they are true in exactly the same interpretations as each other; i.e., both $\mathcal{A} \vDash \mathcal{B}$ and $\mathcal{B} \vDash \mathcal{A}$.

The FOL sentences $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are JOINTLY CONSISTENT IN FOL iff there is some interpretation in which all of the sentences are true. They are JOINTLY INCONSISTENT IN FOL iff there is no such interpretation.

$\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_n \therefore \mathcal{C}$ is VALID IN FOL iff there is no interpretation in which all of the premises are true and the conclusion is false; i.e., $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_n \vDash \mathcal{C}$. It is INVALID IN FOL otherwise.

How does this new idea of *validity in FOL* relate to the original idea of *validity* that we first introduced back in §2? (Recall that an argument is valid iff it is impossible for all of its premises to be true and its conclusion false.) Well, there's some good news!

If an argument is valid in FOL, then it is valid.

Here's why. If $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ is valid in FOL, then there is no interpretation which makes all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ true whilst making $\mathcal{C}$ false. This means that it is *logically impossible* for $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ all to be true whilst $\mathcal{C}$ is false. But this is just what it takes for $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ to be valid!

So every argument that is valid in FOL is valid *full stop*. But what about the other way around? Well, it is certainly too much to hope that absolutely *every* valid argument will be valid in FOL. As we noted in §4.2, some arguments are non-formally valid (i.e. valid, but not just in virtue of their form). Here is an example:

The ball is green all over.
So: The ball is not red all over.

If we try to symbolise this argument in FOL, we get something like this:

$$\exists x\big[Bx \land \forall y(By \to x = y) \land Gx\big] \therefore \exists x\big[Bx \land \forall y(By \to x = y) \land \neg Rx\big]$$

This argument is not valid in FOL. (We'll talk about how to show that arguments are not valid in FOL in §28.) And this shouldn't come as a surprise: an argument is valid in FOL just in case it has a *form* which guarantees that no interpretation can make its premises true without making its conclusion true too.

So, non-formally valid arguments are not valid in FOL. But what about the *formally* valid arguments (i.e. the arguments which are valid just in virtue of their form)? Is every formally valid argument valid in FOL? That turns out to be a very big open question in the philosophy of logic. There are certainly some valid arguments in natural language that are difficult to symbolise in FOL. Here are a couple of examples:

Grass is green.
So: It is not impossible for grass to be green.

Alice and Sandra are both logicians.
So: Alice and Sandra have something in common.

Some philosophers and logicians have claimed that, if we want to deal with these kinds of arguments, then we'll need to move beyond FOL.[1] However, other philosophers have argued that there are clever ways of accommodating these arguments within FOL itself.[2] But whichever side of that debate we fall on, here's what we can all agree on: FOL is an extremely powerful formal system, which allows us to symbolise a huge variety of subtle arguments from science, mathematics, philosophy and everyday life.

[1] To deal with the first kind of argument, we might move to a *Modal Logic*. To deal with the second, we might move to a *Second-Order Logic*. If you would like to learn more about these logics, take a look at the primers I wrote and posted to my website: http://www.rtrueman.com/forallx.html.

[2] David Lewis argued that we can adequately symbolise the first kind of argument in FOL: David Lewis, *On the Plurality of Worlds*, 1986, Oxford: Blackwell Publishing. Willard Van Orman Quine argued that, if we can make any sense of the second kind of argument at all, then it can by symbolised in FOL: W.V.O. Quine, *Philosophy of Logic*, Cambridge, MA: Harvard University Press, pp. 66-8.

Using interpretations 28

28.1 Truths of FOL

Suppose we want to show that '$\exists x Axx \rightarrow Bd$' is *not* a truth of FOL. 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 '$\exists x Axx \rightarrow Bd$' to be false, the antecedent ('$\exists x Axx$') must be true, and the consequent ('Bd') must be false. To construct such an interpretation, we start by specifying a domain. Keeping the domain small makes it easier to specify what the predicates will be true of, so we shall 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 name something in the domain, so we have no option but:

d: Paris

Recall that we want '$\exists x Axx$' to be true, so we want all members of the domain to be paired with themselves in the extension of 'A'. We can just offer:

A: ______$_1$ is identical with ______$_2$

Now 'Add' is true, so it is surely true that '$\exists x Axx$'. Next, we want 'Bd' to be false, so the referent of 'd' must not be in the extension of 'B'. We might simply offer:

B: ______$_1$ is in Germany

Now we have an interpretation where '$\exists x Axx$' is true, but where 'Bd' is false. So there is an interpretation where '$\exists x Axx \rightarrow Bd$' is false. So '$\exists x Axx \rightarrow Bd$' is not a truth of FOL.

In this example, we indirectly fixed the extension of 'B' by saying that it will be true of exactly the same things as '______$_1$ is in Germany'. But if we wanted, we could have fixed it directly, by just stipulating that 'B' is not true of anything in the domain.

To show that $\mathcal{A}$ is not a logical truth, it suffices to find an interpretation where $\mathcal{A}$ is false.

28.2 Contradictions in FOL

Let's look at another example. This time, we will show that '$\exists x Axx \rightarrow Bd$' is not a contradiction in FOL. We need only specify an interpretation in which '$\exists x Axx \rightarrow Bd$' is true; i.e., an interpretation in which either '$\exists x Axx$' is false or 'Bd' is true. Here is one:

domain: Paris
A: $\_\_\_\_\__1$ is identical with $\_\_\_\_\__2$
B: $\_\_\_\_\__1$ is in France
d: Paris

This shows that there is an interpretation where '$\exists x Axx \rightarrow Bd$' is true. So '$\exists x Axx \rightarrow Bd$' is not a contradiction in FOL.

Again, we chose to fix the extensions of 'A' and 'B' indirectly, but we could fix them directly. Here's an example of how:

domain: Paris
A: <Paris, Paris>
B: Paris

Both '$\exists x Axx$' and 'Bd' are true on this interpretation, and so this interpretation is also enough to show that '$\exists x Axx \rightarrow Bd$' is not a contradiction in FOL. '$\exists x Axx \rightarrow Bd$' is not a contradiction in FOL.

> To show that $\mathcal{A}$ is not a contradiction, it suffices to find an interpretation where $\mathcal{A}$ is true.

28.3 Equivalence in FOL

Suppose we want to show that '$\forall x Sx$' and '$\exists x Sx$' are not equivalent in FOL. 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 shall need at least two objects. (If we chose a domain with only one member, the two sentences would end up with the same truth-value. In order to see why, try constructing some partial interpretations with one-member domains.) For concreteness, let's take:

domain: Ornette Coleman, Miles Davis

We can make '$\exists x Sx$' true by including something in the extension of 'S', and we can make '$\forall x Sx$' false by leaving something out of the extension of 'S'. For concreteness we shall offer:

S: $\_\_\_\_\__1$ plays saxophone

Now '$\exists x Sx$' is true, because 'S' 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 '$\exists x Sx$' was true in the original interpretation. Similarly, '$\forall x Sx$' is false, because 'S' 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 '$\forall x Sx$' was false in the original interpretation. We have provided a counter-interpretation to the claim that '$\forall x Sx$' and '$\exists x Sx$' are logically equivalent.

> To show that $\mathscr{A}$ and $\mathscr{B}$ are not equivalent in FOL, it suffices to find an interpretation where one is true and the other is false.

28.4 Consistency in FOL

Suppose we wanted to show that these sentences are jointly consistent in FOL:

$$\exists x(Gx \to Ga) \therefore \exists x Gx \land \neg Ga$$

We need to show that there is an interpretation which makes both sentences true. The second sentence is a conjunction, so to make it true, we need to make both of its conjuncts true. Clearly, our domain must contain two objects. Let's try:

domain: Karl Marx, Ludwig von Mises
Gx: $\_\_\_\_\__1$ 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 '$\exists x Gx \land \neg Ga$' is true, as required.

But does this interpretation make the first sentence true? Yes it does! Note that '$Ga \to Ga$' is true. (Indeed, it is a truth of FOL.) But then certainly '$\exists x(Gx \to Ga)$' is true. So this interpretation makes the first sentence true as well as the second. It follows that these sentences are jointly consistent in FOL.

> To show that $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are jointly consistent in FOL, it suffices to find an interpretation where all of $\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_n$ are true.

28.5 Validity in FOL

Now let's imagine that we want to demonstrate that this argument is *not* valid in FOL:

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

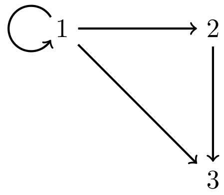
What we need to do is cook up an interpretation that makes the premises true and the conclusion false. Here is a suggestion:

domain: Married people
L: $\_\_\_\_\__1$ is married to $\_\_\_\_\__y$

The premise is clearly true on this interpretation: each person in the domain is married to someone in the domain. Hence '$\forall x \exists y Lxy$' is true. But the conclusion is clearly false, for that would require that there is some single person who is married to everyone in the domain, and there is no such person. So the argument is invalid in FOL.

It is also worth noting that this interpretation also demonstrates that '$\forall x \exists y Lxy$' and '$\neg \exists y \forall x Lxy$' are jointly consistent in FOL.

For the third example, let's mix things up a bit. In §25, we described how to 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
- $\exists x \exists y \exists z (\neg y = z \land Rxy \land Rzx)$ — witness 2
- $\exists x \forall y \neg Rxy$ — witness 3
- $\exists x (\exists y Ryx \land \neg \exists y Rxy)$ — witness 3

This immediately shows that all of the preceding six sentences are jointly consistent in FOL. And that means that this interpretation also shows us that the following arguments are invalid in FOL:

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

and many more besides.

> To show that $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ is invalid in FOL, it suffices to find an interpretation where all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are true and where $\mathcal{C}$ is false.
>
> That same interpretation will show that $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n, \neg\mathcal{C}$ are jointly consistent.

When you provide an interpretation to refute a claim—e.g. when you provide an interpretation to refute the claim that an argument is valid in FOL—this is sometimes called providing a *counter-interpretation* (or *counter-model*) to that claim.

Practice exercises

A. Show that each of the following is neither a truth of FOL nor a contradiction in FOL:

1. $Da \land Db$
2. $\exists x Txh$
3. $Pm \land \neg \forall x Px$
4. $\forall z Jz \leftrightarrow \exists y Jy$
5. $\forall x (Wxmn \lor \exists y Lxy)$
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 equivalent in FOL:

1. Ja, Ka
2. $\exists xJx$, Jm
3. $\forall xRxx$, $\exists xRxx$
4. $\exists xPx \rightarrow Qc$, $\exists x(Px \rightarrow Qc)$
5. $\forall x(Px \rightarrow \neg Qx)$, $\exists x(Px \wedge \neg Qx)$
6. $\exists x(Px \wedge Qx)$, $\exists x(Px \rightarrow Qx)$
7. $\forall x(Px \rightarrow Qx)$, $\forall x(Px \wedge Qx)$
8. $\forall x\exists yRxy$, $\exists x\forall yRxy$
9. $\forall x\exists yRxy$, $\forall x\exists yRyx$

C. Show that the following sentences are jointly consistent in FOL:

1. $Ma, \neg Na, Pa, \neg Qa$
2. $Lee, Leg, \neg Lge, \neg Lgg$
3. $\neg(Ma \wedge \exists xAx), Ma \vee Fa, \forall x(Fx \rightarrow Ax)$
4. $Ma \vee Mb, Ma \rightarrow \forall x\neg Mx$
5. $\forall yGy, \forall x(Gx \rightarrow Hx), \exists y\neg Iy$
6. $\exists x(Bx \vee Ax), \forall x\neg Cx, \forall x\big[(Ax \wedge Bx) \rightarrow Cx\big]$
7. $\exists xXx, \exists xYx, \forall x(Xx \leftrightarrow \neg Yx)$
8. $\forall x(Px \vee Qx), \exists x\neg(Qx \wedge Px)$
9. $\exists z(Nz \wedge Ozz), \forall x\forall y(Oxy \rightarrow Oyx)$
10. $\neg\exists x\forall yRxy, \forall x\exists yRxy$
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 in FOL:

1. $\forall x(Ax \rightarrow Bx) \therefore \exists xBx$
2. $\forall x(Rx \rightarrow Dx), \forall x(Rx \rightarrow Fx) \therefore \exists x(Dx \wedge Fx)$
3. $\exists x(Px \rightarrow Qx) \therefore \exists xPx$
4. $Na \wedge Nb \wedge Nc \therefore \forall xNx$
5. $Rde, \exists xRxd \therefore Red$
6. $\exists x(Ex \wedge Fx), \exists xFx \rightarrow \exists xGx \therefore \exists x(Ex \wedge Gx)$
7. $\forall xOxc, \forall xOcx \therefore \forall xOxx$
8. $\exists x(Jx \wedge Kx), \exists x\neg Kx, \exists x\neg Jx \therefore \exists x(\neg Jx \wedge \neg Kx)$
9. $Lab \rightarrow \forall xLxb, \exists xLxb \therefore Lbb$
10. $\forall x(Dx \rightarrow \exists yTyx) \therefore \exists y\exists z\ \neg y = z$

29 Reasoning about all interpretations

29.1 When a single interpretation won't do

We can show that a sentence is *not* a truth of FOL just by providing one carefully specified interpretation: an interpretation in which the sentence is false. To show that something *is* a truth of FOL, 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 truth of FOL 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!

Something similar holds in other cases too. The following table summarises whether a single (counter-)interpretation suffices, or whether we must reason about all interpretations:

	Yes	**No**
truth of FOL?	all interpretations	one counter-interpretation
contradiction in FOL?	all interpretations	one counter-interpretation
equivalent in FOL?	all interpretations	one counter-interpretation
consistent in FOL?	one interpretation	all interpretations
valid in FOL?	all interpretations	one counter-interpretation

29.2 The challenge...

Sometimes, then, we need to reason about *all* interpretations. But that poses a bit of a problem: with the tools available to you so far, reasoning about all interpretations is a serious challenge!

This marks a big difference between TFL and FOL. Where FOL requires *interpretations*, TFL only required *valuations* (i.e. arbitrary assignments of truth-values to atoms). There are only finitely many possible valuations of finitely many atoms, and we can neatly arrange those valuations into truth-tables. By contrast, there are infinitely many interpretations for any given sentence(s), and so reasoning about all interpretations can be a deeply tricky business.

Let's look at an example, to get a sense of just how tricky it can be. Here is an argument which is obviously valid:

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

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

> Consider an arbitrary interpretation in which the premise '$\forall x(Hx \land Jx)$' is true. It follows that, however we expand the interpretation with a new name, for example 'c', '$Hc \land Jc$' 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 '$\forall x Hx$' is true in the old interpretation. And we assumed nothing about the interpretation except that it was one in which '$\forall x(Hx \land Jx)$' is true. So any interpretation in which '$\forall x(Hx \land Jx)$' is true is one in which '$\forall x Hx$' is true. Thus the argument is valid in FOL!

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

In fact, this is so torturous that we need to think of a better way of doing things. And now for the good news: we already know a better way! In Chapter 4, you learned how to construct proofs for arguments written in TFL. If we take that proof-system, and add a few of extra rules for the quantifiers and identity, then we'll have a proof-system for FOL. The aim of the next chapter is just to spell out those extra rules.

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 chapter 4. We shall also use the proof-theoretic notions (particularly, the symbol '$\vdash$') introduced in that chapter. However, we will also need some new basic rules to govern the quantifiers, and to govern the identity sign.

We'll start in this section with the basic rules for the quantifiers. Like every other logical operator, the quantifiers are governed by two kinds of rule: *introduction* rules, and *elimination* rules. Unfortunately, each quantifier has an easy rule, and a *hard* rule. Let's begin with the easy rules!

30.1 Universal elimination

From the claim that everything is F, you can infer that any particular thing is F. You name it; it's F. So the following should be fine:

$$\begin{array}{r|ll}
1 & \forall x Rxxd & \\
\hline
2 & Raad & \forall\text{E } 1
\end{array}$$

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

$$\begin{array}{r|ll}
1 & \forall x Rxxd & \\
\hline
2 & Rddd & \forall\text{E } 1
\end{array}$$

We obtained line 2 here by dropping the universal quantifier and replacing every instance of 'x' with 'd'. We could have done the same with any other name we wanted.

This motivates the UNIVERSAL ELIMINATION rule ($\forall$E):

$$\begin{array}{r|ll}
m & \forall \mathscr{x}\mathscr{A}(\ldots \mathscr{x} \ldots \mathscr{x} \ldots) & \\
 & \mathscr{A}(\ldots \mathscr{c} \ldots \mathscr{c} \ldots) & \forall\text{E } m
\end{array}$$

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.

It should be emphasised 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	$Bb \to Bk$	naughty attempt to invoke $\forall$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 §20.)

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	$\exists x Raax$	$\exists$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	$\exists x Rxxd$	$\exists$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	$\exists x Rxad$	$\exists$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 shall need to introduce some new notation.

If $\mathcal{A}$ is a sentence containing the name c, we can display this by writing '$\mathcal{A}(\ldots c \ldots c \ldots)$'. We shall write '$\mathcal{A}(\ldots \chi \ldots c \ldots)$' to indicate any formula obtained by replacing *some or all* of the instances of the name c with the variable χ. Armed with this, our EXISTENTIAL INTRODUCTION rule is:

m	$\mathcal{A}(\ldots c \ldots c \ldots)$	
	$\exists \chi \mathcal{A}(\ldots \chi \ldots c \ldots)$	$\exists$I m

χ must not occur in $\mathcal{A}(\ldots c \ldots c \ldots)$

The constraint is included to guarantee that we don't introduce a variable clash, and thus that we have a genuine sentence of FOL (see §21.4). Thus the following is allowed:

$$
\begin{array}{r|ll}
1 & Raad & \\ \cline{2-2}
2 & \exists x Rxad & \exists\text{I } 1 \\
3 & \exists y \exists x Rxyd & \exists\text{I } 2
\end{array}
$$

But this is banned:

$$
\begin{array}{r|ll}
1 & Raad & \\ \cline{2-2}
2 & \exists x Rxad & \exists\text{I } 1 \\
3 & \exists x \exists x Rxxd & \text{naughty attempt to invoke } \exists\text{I } 2
\end{array}
$$

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

30.3 Universal introduction

That's the easy quantifier rules covered. It's time to start on the *hard* rules, beginning with universal introduction.

Suppose you wanted to prove that the internal angles of a triangle always add up to 180°. How would you do it? Your first thought might be to go through every possible triangle, and show one-by-one that the internal angles of each triangle add up to 180°. Unfortunately, however, that just won't work. There are *infinitely many* possible triangles, and so you'll never finish the job.

Here's a better plan. You could pick an *arbitrary* triangle, show that the internal angles of this *arbitrary* triangle add up to 180°, and so conclude that this is what the internal angles of a triangle always add up to. This method definitely works—it's the sort of thing that real mathematicians actually do—but *how* does it work?

Well, when we call the the triangle that you pick 'arbitrary', we're not saying that there's anything special about that triangle: it's not some special, perfect Platonic form; it's just another ordinary triangle. What's special is the attitude that you are taking to the triangle. When you think of the triangle as arbitrary, you are choosing to ignore everything that distinguishes it from any other triangle. You will not allow yourself to pay attention to any information about this triangle, unless you already know that it applies to every other triangle too. If you can maintain this attitude and *still* show that the internal angles of this arbitrary triangle add up to 180°, then it must follow that the internal angles of *every* triangle add up to 180°.

We can formalise this talk about *arbitrary choices* in our UNIVERSAL INTRODUCTION rule (∀I):

$$\begin{array}{r|ll} m & \mathscr{A}(\ldots c \ldots c \ldots) & \\ & \forall \chi \mathscr{A}(\ldots \chi \ldots \chi \ldots) & \forall\text{I } m \end{array}$$

c must not occur in any assumptions that are undischarged at line m (including premises)
c must not occur in $\forall \chi \mathscr{A}(\ldots \chi \ldots \chi \ldots)$
χ must not occur in $\mathscr{A}(\ldots c \ldots c \ldots)$

The requirement that c must not occur in any undischarged assumptions is what guarantees that our choice of name is *arbitrary* in the relevant sense: if c does not appear in any undischarged assumptions, then we don't have any special information about the object that it stands for; everything we know about that object will equally apply to every other object in the domain.[1]

Let's look at an example of this rule in action. Here is an argument that is valid in FOL:

$$\forall x(Fx \to Gx), \forall y(Gy \to Hy) \therefore \forall z(Fz \to Hz)$$

And here is a proof which vindicates this argument:

$$\begin{array}{r|lll} 1 & \forall x(Fx \to Gx) & & \\ 2 & \underline{\forall y(Gy \to Hy)} & & \\ 3 & \quad \underline{Fa} & & \\ 4 & \quad Fa \to Ga & & \forall\text{E } 1 \\ 5 & \quad Ga & & \to\text{E } 4, 3 \\ 6 & \quad Ga \to Ha & & \forall\text{E } 2 \\ 7 & \quad Ha & & \to\text{E } 6, 5 \\ 8 & Fa \to Ha & & \to\text{I } 3\text{–}7 \\ 9 & \forall z(Fz \to Hz) & & \forall\text{I } 8 \end{array}$$

The proof works like this. We want to use ∀I to infer '$\forall z(Fz \to Hz)$', which means we first need to pick an arbitrary name, 'a', and prove '$Fa \to Ha$'. And when we say that 'a' is an arbitrary name, we mean that it does not appear in any assumptions that are still undischarged by the time we apply ∀I. But note: the restriction is only on *undischarged* assumptions. It's fine that we used 'a' in the assumption at line 3, because that assumption was *discharged* by line 8, which is the line that we applied ∀I to.

When applying ∀I, it is absolutely crucial that we make sure that c is an arbitrary name, which doesn't appear in any undischarged assumptions. To see why this matters, consider this terrible argument:

[1] Recall from §14 that we are treating '⊥' as a canonical contradiction. But if it were a canonical contradiction involving some *name*, it might interfere with the constraint mentioned here. To avoid such problems, we shall treat '⊥' as a canonical contradiction *that involves no particular names*, e.g. '$\forall xFx \wedge \neg\forall xFx$'.

> Everyone loves David Attenborough; therefore everyone loves themselves.

We might symbolise this obviously invalid inference pattern as:

$$\forall x Lxd \therefore \forall x Lxx$$

Now, suppose we tried to offer a proof that vindicates this argument:

1	$\forall x Lxd$	
2	Ldd	$\forall$E 1
3	$\forall x Lxx$	naughty attempt to invoke $\forall$I 2

This is not allowed, because 'd' 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 aren't treating it as an *arbitrary* object.

It is also important to remember that, when we are applying $\forall$I, we have to replace *every* occurrence of c in $\mathscr{A}(\ldots c \ldots c \ldots)$ with χ. We can't just replace some and not others. To see why this matters, consider this terrible argument:

> Everyone is the same age as themselves; therefore everyone is the same age as David Attenborough.

We might symbolise this obviously invalid inference pattern as:

$$\forall x Axx \therefore \forall x Axd$$

Now imagine we tried to vindicate this argument with the following proof:

1	$\forall x Axx$	
2	Add	$\forall$E 1
3	$\forall x Axd$	naughty attempt to invoke $\forall$I 2

This is not allowed, because we didn't replace *all* of the occurrences of 'd' in line 2 with an 'x'.

30.4 Existential elimination

We turn now to the elimination rule for the existential quantifier. I should warn you up front: this is the *hardest* rule of the entire natural deduction system.

Suppose you know that some Philosophy lecturer at York can speak English and German. It clearly follows that some Philosophy lecturer at York can speak German. But how could you *demonstrate* that this follows?

This is a difficult question. The problem is that simply knowing the existential generalisation doesn't tell you *which* lecturer can speak English and German. It could be Hannah Carnegy-Arbuthnott, or Chris Jay, or Louise Richardson, or... And now it looks all but impossible to squeeze any information out of this existential generalisation!

However, there is a sneaky way of getting around this problem. We could reason like this:

> First, just *arbitrarily* pick one of the Philosophy lecturers, let's say Chris, and assume, just for the sake of argument, that *he* can speak English and German. You can now draw all sorts of inferences from your new assumption. Some of those inferences will depend on the fact that you picked Chris in particular, but others won't.
>
> For example, from the assumption that Chris speaks English and German, you can infer that Chris speaks German (via conjunction elimination). This inference *does* depend on the fact that you picked Chris at the outset; if you had picked Hannah instead, then you obviously couldn't have inferred that *Chris* speaks German.
>
> But equally, from the assumption that Chris speaks English and German, you can also infer that *some* Philosophy lecturer at York speaks German (via conjunction elimination followed by existential introduction). And this inference does *not* depend on the fact that you picked Chris at the outset; if you had picked Hannah, or any other Philosophy lecturer at York, you could have inferred just the same thing. So, since you know that *some* philosophy lecturer at York does speak English and German, you can safely infer that some philosophy lecturer at York speaks German.

We might try to capture this reasoning pattern in a proof as follows:

1	$\exists x(Ex \wedge Gx)$	
2	$\quad Ec \wedge Gc$	
3	$\quad Gc$	$\wedge$E 2
4	$\quad \exists xGx$	$\exists$I 3
5	$\exists xGx$	$\exists$E 1, 2–4

Let's break this proof down. We started by writing down our initial premise. At line 2, we made an additional assumption: '$Ec \wedge Gc$'. This was just an arbitrary substitution instance of '$\exists x(Ex \wedge Gx)$'. On this assumption, we established '$\exists xGx$'. But note that we had made no *special* assumptions about the object named by 'c'; we had *only* assumed that it satisfies '$Ex \wedge Gx$'. So 'c' was an arbitrary choice of name. Moreover, the conclusion of our subproof at line 4, '$\exists xGx$', would have followed no matter which arbitrary name we chose. So, since line 1 told us that *something* satisfies '$Ex \wedge Gx$', we can discharge the specific assumption of '$Ec \wedge Gc$' at line 2, and simply infer '$\exists xGx$' in the main proof at line 5.

Putting this together, we obtain the EXISTENTIAL ELIMINATION rule ($\exists$E):

m	$\exists \mathscr{X}\mathscr{A}(\ldots\mathscr{X}\ldots\mathscr{X}\ldots)$	
n	$\quad \mathscr{A}(\ldots\mathscr{c}\ldots\mathscr{c}\ldots)$	
o	$\quad \mathscr{B}$	
	$\mathscr{B}$	$\exists$E m, n–o

$\mathscr{c}$ must not occur in any assumptions that are undischarged at line n (including premises)
$\mathscr{c}$ must not occur in $\exists \mathscr{X}\mathscr{A}(\ldots\mathscr{X}\ldots\mathscr{X}\ldots)$
$\mathscr{c}$ must not occur in $\mathscr{B}$

As with universal introduction, the constraints are extremely important. Let's go through them, one by one.

First, consider this bad argument:

> Chris Jay is a lecturer. There is someone who is not a lecturer. So someone is both a lecturer and not a lecturer.

We might symbolise this obviously invalid inference pattern as follows:

$$Lc, \exists x \neg Lx \therefore \exists x(Lx \land \neg Lx)$$

Now, suppose we tried to offer a proof that vindicates this argument:

1	Lc	
2	$\exists x \neg Lx$	
3	$\quad \neg Lc$	
4	$\quad Lc \land \neg Lc$	$\land$E 1, 3
5	$\quad \exists x(Lx \land \neg Lx)$	$\exists$I 4
6	$\exists x(Lx \land \neg Lx)$	naughty attempt to invoke $\exists$E 2, 3–5

The last line of the proof is not allowed, because it violates the first constraint on $\exists$E: the name that we used in our substitution instance for '$\exists x \neg Lx$' on line 3, namely 'c', occurs in line 4. In other words, 'c' isn't really an *arbitrarily* chosen name in this proof.

Now take a look at this stinker:

> Everyone respects someone; therefore someone respects themself.

We might symbolise this obviously invalid inference pattern as follows:

$$\forall x \exists y Rxy \therefore \exists x Rxx$$

But now imagine that we tried to back up this inference with the following proof:

$$
\begin{array}{rlll}
1 & \forall x \exists y Rxy & & \\
2 & \exists y Rcy & & \\
3 & \quad Rcc & & \\
4 & \quad \exists x Rxx & & \exists\text{I } 3 \\
5 & \exists x Rxx & & \text{naughty attempt to invoke } \exists\text{E } 2, 3\text{–}4
\end{array}
$$

The last line of the proof is not allowed, because it violates the second constraint on ∃E: at line 3, we tried to substitute 'c' for 'y' in '$\exists yRcy$', but 'c' already appears in that existential generalisation. In other words, 'c' *still* isn't arbitrary enough for this ∃E.

One last mess of an argument (and this one should be familiar):

> Someone can speak English and German; so Chris can speak German.

We might symbolise this obviously invalid inference pattern as follows:

$$\exists x(Ex \land Gx) \therefore Gc$$

And here's a bad proof for this argument:

$$
\begin{array}{rlll}
1 & \exists x(Ex \land Gx) & & \\
2 & \quad Ec \land Gc & & \\
3 & \quad Gc & & \land\text{E } 2 \\
4 & Gc & & \text{naughty attempt to invoke } \exists\text{E } 1, 2\text{–}3
\end{array}
$$

The last line of the proof is not allowed, because it violates the third constraint on ∃E: at line 2, we picked the name 'c' to substitute for 'x' in '$\exists x(Ex \land Gx)$'; but that name still appears in 'Gc', and so that sentence cannot be pulled out of the subproof with ∃E. This time, the problem is not that 'c' isn't an arbitrarily chosen name. The problem is that the conclusion we reached wouldn't have followed from *other* arbitrary choices of name.

One way of meeting this third requirement would be to perform ∃I on line 3, which would deliver '$\exists xGx$', and then use ∃E to pull that generalisation out into the main proof. (That is exactly what we did in the proof on page 154.) And this is often the way: the trick in many proofs is to first use ∃I and then use ∃E. But that isn't how *every* proof goes. We can illustrate another common trick by looking at this argument:

$$\forall x(Fx \to Gx) \therefore \neg\exists x(Fx \land \neg Gx)$$

Here's a proof for this argument:

$$
\begin{array}{lll}
1 & \forall x(Fx \to Gx) & \\
2 & \quad \exists x(Fx \land \neg Gx) & \\
3 & \quad\quad Fa \land \neg Ga & \\
4 & \quad\quad Fa & \land\text{E } 3 \\
5 & \quad\quad \neg Ga & \land\text{E } 3 \\
6 & \quad\quad Fa \to Ga & \forall\text{E } 1 \\
7 & \quad\quad Ga & \to\!\text{E } 6, 4 \\
8 & \quad\quad \bot & \bot\text{I } 7, 5 \\
9 & \quad \bot & \exists\text{E } 2, 3\text{–}8 \\
10 & \neg\exists x(Fx \land \neg Gx) & \neg\text{I } 2\text{–}9
\end{array}
$$

This proof meets the third constraint on ∃E, because 'a' doesn't occur in '⊥'. In fact, *no names* occur in '⊥',[2] and so you can always use ∃E to extract a contradiction.

Practice exercises

A. The following two 'proofs' are *incorrect*. Explain why both are incorrect. Also, provide interpretations which show that the corresponding arguments are invalid in FOL:

$$
\begin{array}{lll}
1 & \forall x Rxx & \\
2 & Raa & \forall\text{E } 1 \\
3 & \forall y Ray & \forall\text{I } 2 \\
4 & \forall x \forall y Rxy & \forall\text{I } 3
\end{array}
$$

$$
\begin{array}{lll}
1 & \forall x \exists y Rxy & \\
2 & \exists y Ray & \forall\text{E } 1 \\
3 & \quad Raa & \\
4 & \quad \exists x Rxx & \exists\text{I } 3 \\
5 & \exists x Rxx & \exists\text{E } 2, 3\text{–}4
\end{array}
$$

[2] But recall fn.1: if we choose to read '⊥' as an abbreviation for a canonical contradiction, then we must choose a contradiction that involves no names.

B. The following three proofs are missing their citations (rule and line numbers). Add them, to turn them into bona fide proofs.

1	$\forall x \exists y(Rxy \lor Ryx)$
2	$\forall x \neg Rmx$
3	$\exists y(Rmy \lor Rym)$
4	$\quad Rma \lor Ram$
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	$\quad Ka \land Laa$
10	$\quad \exists x(Kx \land Lxx)$
11	$\exists x(Kx \land Lxx)$

C. In §20 problem part 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 symbolise (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'. Symbolise 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. $\vdash \forall x Fx \vee \neg\forall x Fx$
2. $\vdash \forall z(Pz \vee \neg Pz)$
3. $\forall x(Ax \rightarrow Bx), \exists x Ax \vdash \exists x Bx$
4. $\forall x(Mx \leftrightarrow Nx), Ma \wedge \exists x Rxa \vdash \exists x Nx$
5. $\forall x\forall y Gxy \vdash \exists x Gxx$
6. $\vdash \forall x Rxx \rightarrow \exists x\exists y Rxy$
7. $\vdash \forall y\exists x(Qy \rightarrow Qx)$
8. $Na \rightarrow \forall x(Mx \leftrightarrow Ma), Ma, \neg Mb \vdash \neg Na$
9. $\forall x\forall y(Gxy \rightarrow Gyx) \vdash \forall x\forall y(Gxy \leftrightarrow Gyx)$
10. $\forall x(\neg Mx \vee Ljx), \forall x(Bx \rightarrow Ljx), \forall x(Mx \vee Bx) \vdash \forall x Ljx$

F. Write a symbolisation key for the following argument, symbolise it, and prove it:

> There is someone who likes everyone who likes everyone that she likes. Therefore, there is someone who likes herself.

G. For each of the following pairs of sentences: If they are provably equivalent, give proofs to show this. If they are not, construct an interpretation to show that they are not equivalent in FOL.

1. $\forall x Px \rightarrow Qc, \forall x(Px \rightarrow Qc)$
2. $\forall x\forall y\forall z Bxyz, \forall x Bxxx$
3. $\forall x\forall y Dxy, \forall y\forall x Dxy$
4. $\exists x\forall y Dxy, \forall y\exists x Dxy$
5. $\forall x(Rca \leftrightarrow Rxa), Rca \leftrightarrow \forall x Rxa$

H. For each of the following arguments: If it is valid in FOL, give a proof. If it is invalid in FOL, construct an interpretation to show that it is invalid in FOL.

1. $\exists y\forall x Rxy \therefore \forall x\exists y Rxy$
2. $\exists x(Px \wedge \neg Qx) \therefore \forall x(Px \rightarrow \neg Qx)$
3. $\forall x(Sx \rightarrow Ta), Sd \therefore Ta$
4. $\forall x(Ax \rightarrow Bx), \forall x(Bx \rightarrow Cx) \therefore \forall x(Ax \rightarrow Cx)$
5. $\exists x(Dx \vee Ex), \forall x(Dx \rightarrow Fx) \therefore \exists x(Dx \wedge Fx)$
6. $\forall x\forall y(Rxy \vee Ryx) \therefore Rjj$
7. $\exists x\exists y(Rxy \vee Ryx) \therefore Rjj$
8. $\forall x Px \rightarrow \forall x Qx, \exists x\neg Px \therefore \exists x\neg Qx$

Derived quantifier rules 31

In this section, we shall add some additional rules to the basic rules of the previous section. These govern the interaction of quantifiers and negation.

31.1 Conversion of quantifiers

In §19, we noted that $\neg\exists x\mathcal{A}$ is logically equivalent to $\forall x\neg\mathcal{A}$. We shall add some rules to our proof system that govern this. In particular, we add:

m	$\forall\chi\neg\mathcal{A}$	
	$\neg\exists\chi\mathcal{A}$	CQ m

and

m	$\neg\exists\chi\mathcal{A}$	
	$\forall\chi\neg\mathcal{A}$	CQ m

Equally, we add:

m	$\exists\chi\neg\mathcal{A}$	
	$\neg\forall\chi\mathcal{A}$	CQ m

and

m	$\neg\forall\chi\mathcal{A}$	
	$\exists\chi\neg\mathcal{A}$	CQ m

31.2 Deriving the conversion rules

The rules for converting quantifiers that we just introduced do not add any power to our proof system for FOL; they just make it easier to use. We know that they do not add any power, because they can be derived from the basic rules.

Here is a justification for the first CQ rule:

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

And here is a justification of the second CQ rule:

1	$\exists x \neg Ax$	
2	$\mid\ \forall x Ax$	
3	$\mid\ \mid\ \neg Ac$	
4	$\mid\ \mid\ Ac$	$\forall$E 2
5	$\mid\ \mid\ \bot$	$\bot$I 4, 3
6	$\mid\ \bot$	$\exists$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 third and fourth CQ rules as derived rules.

B. Show that the following are jointly contrary:

1. $Sa \to Tm, Tm \to Sa, Tm \land \neg Sa$
2. $\neg \exists x Rxa, \forall x \forall y Ryx$
3. $\neg \exists x \exists y Lxy, Laa$
4. $\forall x(Px \to Qx), \forall z(Pz \to Rz), \forall y Py, \neg Qa \land \neg Rb$

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

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

D. In §20, we considered what happens when we move quantifiers 'across' various logical operators. Show that each pair of sentences is provably equivalent:

1. $\forall x(Fx \land Ga), \forall xFx \land Ga$
2. $\exists x(Fx \lor Ga), \exists xFx \lor Ga$
3. $\forall x(Ga \rightarrow Fx), Ga \rightarrow \forall xFx$
4. $\forall x(Fx \rightarrow Ga), \exists xFx \rightarrow Ga$
5. $\exists x(Ga \rightarrow Fx), Ga \rightarrow \exists xFx$
6. $\exists x(Fx \rightarrow Ga), \forall xFx \rightarrow Ga$

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.

Rules for identity 32

We come now to the very last set of rules we're going to add to our natural deduction system: the rules governing identity. Let's start with the introduction rule.

What does it take for object x to be *numerically identical* to object y? That is a philosophically controversial question! Some metaphysicians subscribe to a principle called the IDENTITY OF INDISCERNIBLES: if x and y are indiscernible in every way, then x is numerically identical to y. However, plenty of philosophers reject this principle. Here's a famous counterexample, due to Max Black: imagine a world that contains nothing but two perfect spheres, both with a radius of 1 mile and both made of pure iron; these spheres seem to be completely indistinguishable, and yet there are *two* of them, not just one.[1]

Maybe you're convinced by this kind of counterexample, maybe you're not. But we don't want to build anything too metaphysically controversial into our proof system. So we won't assume the identity of indiscernibles in our system. It follows that, no matter how much you tell me about x and y, you cannot use our proof system to prove that x is identical to y. (Unless, of course, what you tell me is that x is identical to y. But then the proof will hardly be very illuminating.)

But don't despair! Logic always guarantees that every object is identical *to itself*. No premises, then, are required in order to conclude $\mathcal{a} = \mathcal{a}$. So this will be our IDENTITY INTRODUCTION rule:

$$\big|\ \mathcal{a} = \mathcal{a} \qquad =\text{I}$$

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

So far, identity has been pretty boring. But our elimination rule is much more fun. Although the identity of indiscernibles is highly controversial, the INDISCERNIBILITY OF IDENTICALS is a logical guarantee: if x is identical to y, then x and y must be indiscernible in every way. (This is also called LEIBNIZ'S LAW).

Here's how we can incorporate the indiscernibility of identicals into our proof system. Suppose you have established '$a = b$'. In that case, you can take any sentence with the name 'a' in it, and replace some or all of the occurrences of 'a' with the name 'b'. For example, from 'Raa' and '$a = b$', you are justified in inferring 'Rab', 'Rba' or 'Rbb'. More generally:

[1]Max Black, 'The Identity of Indiscernibles', 1952, *Mind* 61, pp. 153–64

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 $\exists$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

Let's take a look at these rules in action. Here's an argument that is valid in FOL:

$$\forall x(x = a \to Fx), b = a \therefore Fb$$

And here's a proof for this argument:

1	$\forall x(x = a \to Fx)$	
2	$b = a$	
3	$a = a \to Fa$	$\forall$E 1
4	$a = a$	=I
5	Fa	$\to$E 3, 4
6	Fb	=E 2, 5

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. $\exists x Jx, \exists x \neg Jx \vdash \exists x \exists y\ \neg x = y$
7. $\forall x(x = n \leftrightarrow Mx), \forall x(Ox \lor \neg Mx) \vdash On$

8. $\exists x Dx, \forall x(x = p \leftrightarrow Dx) \vdash Dp$
9. $\exists x\big[(Kx \land \forall y(Ky \to x = y)) \land Bx\big], Kd \vdash Bd$
10. $\vdash Pa \to \forall x(Px \lor \neg x = a)$

B. Identity is an EQUIVALENCE RELATION, which means that it is reflexive, symmetric, and transitive:

> REFLEXIVITY: $\forall x\ x = x$
> SYMMETRY: $\forall x \forall y(x = y \to y = x)$
> TRANSITIVITY: $\forall x \forall y \forall z((x = y \land y = z) \to x = z)$

Show that the reflexivity, symmetry and transitivity of identity are all theorems of FOL.

C. Show that the following are provably equivalent:

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

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

D. In §22, we said that the following are logically equivalent symbolisations of the English sentence 'there is exactly one F':

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

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.)

E. Symbolise 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.

Proof-theoretic concepts and semantic concepts 33

We have used two different turnstiles in this book. This,

$$\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{C}$$

means that there is some proof which starts with assumptions $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ and ends with $\mathcal{C}$ (without relying on any additional undischarged assumptions). This is a *proof-theoretic notion.*

By contrast, this,

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

means that there is no valuation (in TFL) or interpretation (in FOL) which makes all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ true and makes $\mathcal{C}$ false. This concerns assignments of truth and falsity to sentences. It is a *semantic notion.*

It is impossible to over-emphasise how different these two notions are. But let's try anyway: *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 the truths of FOL and the theorems of FOL.

To show that a sentence is a theorem, you need only provide a proof. Granted, it may be hard to come up with that proof, but once you have a proof in front of you, it is not so hard to check each line to make sure 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 truth of FOL, though, requires reasoning about all possible interpretations. Given a choice between showing that a sentence is a theorem and showing that it is a truth of FOL, it would be easier to show that it is a theorem.

On the other hand, to show that a sentence is *not* a theorem is hard. We would need to reason about all (possible) proofs. That is very difficult. But to show that a sentence is not a truth of FOL, 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 truth of FOL, it would be easier to show that it is not a truth of FOL.

Fortunately, *a sentence is a theorem of FOL if and only if it is a truth of FOL.* As a result, if we provide a proof of $\mathcal{A}$ on no assumptions, and thus show that $\mathcal{A}$ is a theorem, we can legitimately infer that $\mathcal{A}$ is a truth of FOL; i.e., $\vDash \mathcal{A}$. Similarly, if we construct an interpretation in which $\mathcal{A}$ is false and thus show that it is not a truth of FOL, it follows that $\mathcal{A}$ is not a theorem.

More generally, we have the following powerful results:

SOUNDNESS: If $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{C}$, then $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{C}$
COMPLETENESS: If $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{C}$, then $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{C}$

This shows that, whilst provability and entailment are *different* notions, they are extensionally equivalent. As such:[1]

- An argument is *valid in FOL* iff *the conclusion can be proved from the premises.*
- Two sentences are *equivalent in FOL* iff they are *provably equivalent.*
- Sentences are *jointly consistent in FOL* iff they are *not jointly contrary.*

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 do not be fooled by the similarity of the symbols '$\vDash$' and '$\vdash$'. These two symbols have very different meanings. And 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 we leave introductory logic behind. (It's not *too* hard when we're dealing just with TFL; but it took the genius of Gödel to first demonstrate this for FOL.) We won't cover those results here, but for anyone who is interested, I would highly recommend the classic textbook *Computability and Logic.*[2]

[1]We have similar results for TFL: an argument is tautologically valid iff the conclusion can be proved from the premises just using the TFL rules; two sentences are tautologically equivalent iff they are provably equivalent just using the TFL rules; sentences are jointly tautologically consistent iff '$\bot$' cannot be proved from them just using the TFL rules.

[2]There are several editions of *Computability and Logic*, and all of them are great. But here's the most recent version: George S. Boolos, John P. Burgess, and Richard C. Jeffrey, *Computability and Logic*, 5th edition, 2007, Cambridge: Cambridge University Press.

	Yes	**No**
Is $\mathcal{A}$ a **truth of FOL**?	give a proof which shows $\vdash \mathcal{A}$	give an interpretation in which $\mathcal{A}$ is false
Is $\mathcal{A}$ a **contradiction in FOL**?	give a proof which shows $\vdash \neg\mathcal{A}$	give an interpretation in which $\mathcal{A}$ is true
Are $\mathcal{A}$ and $\mathcal{B}$ **equivalent in FOL**?	give two proofs, one for $\mathcal{A} \vdash \mathcal{B}$ and one for $\mathcal{B} \vdash \mathcal{A}$	give an interpretation in which $\mathcal{A}$ and $\mathcal{B}$ have different truth values
Are $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ **jointly consistent in FOL**?	give an interpretation in which all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are true	prove a contradiction from assumptions $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$
Is $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ **valid in FOL**?	give a proof with assumptions $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ and concluding with $\mathcal{C}$	give an interpretation in which each of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ is true and $\mathcal{C}$ is false

Appendices

Symbolic notation A

A.1 Alternative nomenclature

Truth-functional logic. TFL goes by other names. Sometimes it is called *Sentence 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 emphasise 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. We have stuck with *First-Order Logic* because there are other formal systems which involve predication and quantification. Most notably, there is *Second-Order Logic*, which you can read about here: http://www.rtrueman.com/forallx.html.

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*.

n-place predicates. We have called predicates 'one-place', 'two-place', 'three-place', etc. Other texts respectively call them 'monadic', 'dyadic', 'triadic', etc. Still other texts call them 'unary', 'binary', 'trinary', 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'.

A.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 recognise them if you encounter them in an article or in another book.

Negation. Two commonly used symbols are the *corner*, '$\neg$', and the *tilde*, '$\sim$.' 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 '$\sim$'. 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 '*&*'. Fun fact: the ampersand was reportedly invented by Cicero's slave Tiro.) Using this symbol is not recommended, since it is commonly used in natural English writing (e.g. 'Smith & Sons'). As a symbol in a formal system, the ampersand is not the English word '&', so it is much neater to use a completely different symbol. 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*, '$\supset$'. It is better to use '$\rightarrow$', because '$\supset$' is standardly used in a branch of mathematics called *set theory* to express *proper supersethood*. And to make things even more confusing, proper supersethood is defined in terms of the conditional, like this:

$$x \supset y \leftrightarrow \big[\forall z(z \in y \rightarrow z \in x) \wedge \neg\forall z(z \in x \rightarrow z \in y)\big]$$

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. We prefer the double-headed arrow because it makes it clear that the biconditional is just two conditionals, one in each direction.

Quantifiers. The universal quantifier is typically symbolised 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 '$(x)Fx$' where we would write '$\forall x Fx$'. In some old texts, a capital pi is used for the universal quantifier, '$\Pi x Fx$', and a capital sigma is used for the existential quantifier, '$\Sigma x Fx$'. However, in modern texts, these Greek letters are sometimes used for *substitutional quantifiers*, which are a little different from the kind of quantifier you have learned about in this textbook.

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), Πx
existential quantifier	$\exists x$, Σx

Alternative proof systems B

In formulating our natural deduction system, we have 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 can illustrate this point by considering some alternative treatments of disjunction, negation, and the quantifiers. I shall also explain why I have made the choices that I have.

B.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:

$$
\begin{array}{lll}
m & \mathcal{A} \lor \mathcal{B} & \\
i & \quad \mid \mathcal{A} & \\
j & \quad \mid \mathcal{C} & \\
k & \quad \mid \mathcal{B} & \\
l & \quad \mid \mathcal{C} & \\
n & \mathcal{A} \to \mathcal{C} & \to\text{I } i\text{–}j \\
n+1 & \mathcal{B} \to \mathcal{C} & \to\text{I } k\text{–}l \\
n+2 & \quad \mid \mathcal{C} & \\
n+3 & \quad \mid \mathcal{C} & \text{R } n+2 \\
n+4 & \quad \mid \neg\mathcal{C} & \\
n+5 & \quad \mid \quad \mid \mathcal{A} & \\
n+6 & \quad \mid \quad \mid \mathcal{C} & \to\text{E } n,\ n+5 \\
n+7 & \quad \mid \quad \mid \bot & \bot\text{I } n+6,\ n+4 \\
n+8 & \quad \mid \neg\mathcal{A} & \neg\text{I } n+5\text{–}n+7 \\
n+9 & \quad \mid \mathcal{B} & \text{DS } m,\ n+8 \\
n+10 & \quad \mid \mathcal{C} & \to\text{E } n+1,\ n+9 \\
n+11 & \mathcal{C} & \text{TND } n+2\text{–}n+3,\ n+4\text{–}n+10
\end{array}
$$

So why choose to take ∨E as basic, rather than DS?[1] That was Tim Button's choice, and his reasoning was that DS involves the use of '¬' in the statement of the rule. It is in some sense 'cleaner' for our disjunction elimination rule to avoid mentioning *other* connectives. (Some philosophers, most notably the *intuitionists*, take this sort of thing *very seriously.*)

B.2 Alternative negation rules

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

$$
\begin{array}{lll}
m & \quad \mid \mathcal{A} & \\
n-1 & \quad \mid \mathcal{B} & \\
n & \quad \mid \neg\mathcal{B} & \\
 & \neg\mathcal{A} & \neg\text{I* } m\text{–}n
\end{array}
$$

and the following as their basic negation elimination rule:

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

$$\begin{array}{l|l|ll}
m & & \neg\mathcal{A} & \\
n-1 & & \mathcal{B} & \\
n & & \neg\mathcal{B} & \\
 & \mathcal{A} & & \neg\text{E}^* \; m\text{–}n
\end{array}$$

Using these two rules, we could have derived all of the rules governing negation and contradiction that we have taken as basic (i.e. ⊥I, ⊥E, ¬I and TND). Indeed, we could have avoided all use of the symbol '⊥' 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 choose to separate out contradiction, and to use an explicit rule TND? Again, this was Tim Button's choice, and here was his reasoning.[2]

First: adding the symbol '⊥' to our natural deduction system makes proofs considerably easier to work with.

Second: a lot of fascinating philosophical discussion has focussed on the acceptability or otherwise of *tertium non datur* (i.e. TND) and *ex falso quodlibet* (i.e. ⊥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.

B.3 Alternative quantification rules

An alternative approach to the quantifiers is to take as basic the rules for ∀I and ∀E from §30, and also two CQ rule which allow us to move from $\forall\mathcal{X}\neg\mathcal{A}$ to $\neg\exists\mathcal{X}\mathcal{A}$ and vice versa.[3]

Taking only these rules as basic, we could have derived the ∃I and ∃E rules provided in §30. To derive the ∃I rule is fairly simple. Suppose $\mathcal{A}$ contains the name c, and contains no instances of the variable $\mathcal{X}$, and that we want to do the following:

$$\begin{array}{l|l}
m & \mathcal{A}(\ldots c \ldots c \ldots) \\
k & \exists\mathcal{X}\mathcal{A}(\ldots \mathcal{X} \ldots c \ldots)
\end{array}$$

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

[2]And also 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.

$$
\begin{array}{lll}
m & \mathcal{A}(\ldots c \ldots c \ldots) & \\
m+1 & \quad \mid\ \neg\exists\chi\mathcal{A}(\ldots \chi \ldots c \ldots) & \\
m+2 & \quad \mid\ \forall\chi\neg\mathcal{A}(\ldots \chi \ldots c \ldots) & \text{CQ } m+1 \\
m+3 & \quad \mid\ \neg\mathcal{A}(\ldots c \ldots c \ldots) & \forall\text{E } m+2 \\
m+4 & \quad \mid\ \bot & \bot\text{I } m,\ m+3 \\
m+5 & \neg\neg\exists\chi\mathcal{A}(\ldots \chi \ldots c \ldots) & \neg\text{I } m+1\text{–}m+4 \\
m+6 & \exists\chi\mathcal{A}(\ldots \chi \ldots c \ldots) & \text{DNE } m+5
\end{array}
$$

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,

$$
\begin{array}{ll}
m & \exists\chi\mathcal{A}(\ldots \chi \ldots \chi \ldots) \\
i & \quad \mid\ \mathcal{A}(\ldots c \ldots c \ldots) \\
j & \quad \mid\ \mathcal{B} \\
k & \mathcal{B}
\end{array}
$$

where c does not occur in any undischarged assumptions, or in $\mathcal{B}$, or in $\exists\chi\mathcal{A}(\ldots \chi \ldots \chi \ldots)$. 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:

$$
\begin{array}{lll}
m & \exists\chi\mathcal{A}(\ldots \chi \ldots \chi \ldots) & \\
i & \quad \mid\ \mathcal{A}(\ldots c \ldots c \ldots) & \\
j & \quad \mid\ \mathcal{B} & \\
k & \mathcal{A}(\ldots c \ldots c \ldots) \to \mathcal{B} & \to\text{I } i\text{–}j \\
k+1 & \quad \mid\ \neg\mathcal{B} & \\
k+2 & \quad \mid\ \neg\mathcal{A}(\ldots c \ldots c \ldots) & \text{MT } k,\ k+1 \\
k+3 & \quad \mid\ \forall\chi\neg\mathcal{A}(\ldots \chi \ldots \chi \ldots) & \forall\text{I } k+2 \\
k+4 & \quad \mid\ \neg\exists\chi\mathcal{A}(\ldots \chi \ldots \chi \ldots) & \text{CQ } k+3 \\
k+5 & \quad \mid\ \bot & \bot\text{I } m,\ k+4 \\
k+6 & \neg\neg\mathcal{B} & \neg\text{I } k+1\text{–}k+5 \\
k+7 & \mathcal{B} & \text{DNE } k+6
\end{array}
$$

We are permitted to use $\forall$I on line $k+3$ because c does not occur in any undischarged assumptions or in $\mathcal{B}$. The entries on lines k+4 and k+1 contradict each other, because c does not occur in $\exists\chi\mathcal{A}(\ldots \chi \ldots \chi \ldots)$.

Armed with these derived rules, we could now go on to derive the two remaining CQ rules, exactly as in §31.2.

So, why did we start with all of the quantifier rules as basic, and then derive the CQ rules?

First: 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.

Second: In the derivations of the rules of ∃I and ∃E that were offered in this section, we have invoked DNE. This is a derived rule, whose derivation essentially depends upon the use of TND. But, as was mentioned earlier, TND is a contentious rule. So, if we want to move to a system which abandons TND, but which still allows us to use existential quantifiers, we shall 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 TND, we shall be *unable* to derive the CQ rule which moves from $\neg\forall\chi\mathcal{A}$ to $\exists\chi\neg\mathcal{A}$.)

Quick reference C

C.1 Characteristic Truth Tables

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

$\mathscr{A}$	$\mathscr{B}$	$\mathscr{A}\wedge\mathscr{B}$	$\mathscr{A}\vee\mathscr{B}$	$\mathscr{A}\to\mathscr{B}$	$\mathscr{A}\leftrightarrow\mathscr{B}$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

C.2 Symbolisation

SENTENTIAL CONNECTIVES

It is not the case that P	$\neg P$
Either P, or Q	$(P \vee Q)$
Neither P, nor Q	$\neg(P \vee Q)$ or $(\neg P \wedge \neg Q)$
Both P, and Q	$(P \wedge Q)$
If P, then Q	$(P \to Q)$
P only if Q	$(P \to Q)$
P if and only if Q	$(P \leftrightarrow Q)$
P unless Q	$(P \vee Q)$

PREDICATES

All Fs are Gs	$\forall x(Fx \to Gx)$
Some Fs are Gs	$\exists x(Fx \wedge Gx)$
Not all Fs are Gs	$\neg\forall x(Fx \to Gx)$ or $\exists x(Fx \wedge \neg Gx)$
No Fs are Gs	$\forall x(Fx \to \neg Gx)$ or $\neg\exists x(Fx \wedge Gx)$

IDENTITY

Everything besides c is G	$\forall x(\neg x = c \to Gx)$
Nothing besides c is G	$\forall x(Gx \to x = c)$
c is the G	$\forall x(Gx \leftrightarrow x = c)$
The F is G	$\exists x(Fx \wedge \forall y(Fy \to x = y) \wedge Gx)$
It is not the case that the F is G	$\neg\exists x(Fx \wedge \forall y(Fy \to x = y) \wedge Gx)$
The F is not G	$\exists x(Fx \wedge \forall y(Fy \to x = y) \wedge \neg Gx)$

C.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 symbolisation for 'there are at least $n + 1$ Fs'. Equivalently, we can offer:

one: $\forall x_1 \forall x_2 \big[(Fx_1 \wedge Fx_2) \to x_1 = x_2\big]$
two: $\forall x_1 \forall x_2 \forall x_3 \big[(Fx_1 \wedge Fx_2 \wedge Fx_3) \to (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3)\big]$
three: $\forall x_1 \forall x_2 \forall x_3 \forall x_4 \big[(Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge Fx_4) \to$
$(x_1 = x_2 \vee x_1 = x_3 \vee x_1 = x_4 \vee x_2 = x_3 \vee x_2 = x_4 \vee x_3 = x_4)\big]$
n: $\forall x_1 \ldots \forall x_{n+1} \big[(Fx_1 \wedge \ldots \wedge Fx_{n+1}) \to (x_1 = x_2 \vee \ldots \vee x_n = x_{n+1})\big]$

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: $\exists x \big[Fx \wedge \forall y (Fy \to x = y)\big]$
two: $\exists x_1 \exists x_2 \big[Fx_1 \wedge Fx_2 \wedge \neg x_1 = x_2 \wedge \forall y \big(Fy \to (y = x_1 \vee y = x_2)\big)\big]$
three: $\exists x_1 \exists x_2 \exists x_3 \big[Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge \neg x_1 = x_2 \wedge \neg x_1 = x_3 \wedge \neg x_2 = x_3 \wedge$
$\forall y \big(Fy \to (y = x_1 \vee y = x_2 \vee y = x_3)\big)\big]$
n: $\exists x_1 \ldots \exists x_n \big[Fx_1 \wedge \ldots \wedge Fx_n \wedge \neg x_1 = x_2 \wedge \ldots \wedge \neg x_{n-1} = x_n \wedge$
$\forall y \big(Fy \to (y = x_1 \vee \ldots \vee y = x_n)\big)\big]$

C.4 Basic deduction rules for TFL

Conjunction

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

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

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

Conditional

$$
\begin{array}{r|ll}
i & \quad \mid \underline{\mathcal{A}} & \\
j & \quad \mid \mathcal{B} & \\
 & \mathcal{A} \to \mathcal{B} & \to\text{I } i\text{–}j
\end{array}
$$

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

Contradiction

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

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

Negation

$$
\begin{array}{r|ll}
i & \quad \mid \underline{\mathcal{A}} & \\
j & \quad \mid \bot & \\
 & \neg\mathcal{A} & \neg\text{I } i\text{–}j
\end{array}
$$

Tertium non datur

$$
\begin{array}{r|ll}
i & \quad \mid \underline{\mathcal{A}} & \\
j & \quad \mid \mathcal{B} & \\
k & \quad \mid \underline{\neg\mathcal{A}} & \\
l & \quad \mid \mathcal{B} & \\
 & \mathcal{B} & \text{TND } i\text{–}j, k\text{–}l
\end{array}
$$

Disjunction

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

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

$$
\begin{array}{r|ll}
m & \mathcal{A} \vee \mathcal{B} & \\
i & \quad \mid \underline{\mathcal{A}} & \\
j & \quad \mid \mathcal{C} & \\
k & \quad \mid \underline{\mathcal{B}} & \\
l & \quad \mid \mathcal{C} & \\
 & \mathcal{C} & \vee\text{E } m, i\text{–}j, k\text{–}l
\end{array}
$$

Biconditional

$$
\begin{array}{r|ll}
i & \quad \mid \underline{\mathcal{A}} & \\
j & \quad \mid \mathcal{B} & \\
k & \quad \mid \underline{\mathcal{B}} & \\
l & \quad \mid \mathcal{A} & \\
 & \mathcal{A} \leftrightarrow \mathcal{B} & \leftrightarrow\text{I } i\text{–}j, k\text{–}l
\end{array}
$$

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

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

C.5 Derived rules for TFL

Disjunctive syllogism

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{A}$	
	$\mathcal{B}$	DS m, n

m	$\mathcal{A} \vee \mathcal{B}$	
n	$\neg\mathcal{B}$	
	$\mathcal{A}$	DS m, n

Reiteration

m	$\mathcal{A}$	
	$\mathcal{A}$	R m

Modus Tollens

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

Double-negation elimination

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

De Morgan Rules

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

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

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

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

C.6 Basic deduction rules for FOL

Universal elimination

$$\begin{array}{r|ll} m & \forall \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots) & \\ & \mathcal{A}(\ldots c \ldots c \ldots) & \forall\text{E } m \end{array}$$

Universal introduction

$$\begin{array}{r|ll} m & \mathcal{A}(\ldots c \ldots c \ldots) & \\ & \forall \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots) & \forall\text{I } m \end{array}$$

c must not occur in any assumptions that are undischarged at line m
χ must not occur in $\mathcal{A}(\ldots c \ldots c \ldots)$

Existential introduction

$$\begin{array}{r|ll} m & \mathcal{A}(\ldots c \ldots c \ldots) & \\ & \exists \chi \mathcal{A}(\ldots \chi \ldots c \ldots) & \exists\text{I } m \end{array}$$

χ must not occur in $\mathcal{A}(\ldots c \ldots c \ldots)$

Existential elimination

$$\begin{array}{r|l|ll} m & \exists \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots) & & \\ n & & \mathcal{A}(\ldots c \ldots c \ldots) & \\ \cline{3-3} o & & \mathcal{B} & \\ & \mathcal{B} & & \exists\text{E } m,\ n\text{–}o \end{array}$$

c must not occur in any assumptions that are undischarged at line n, in $\exists \chi \mathcal{A}(\ldots \chi \ldots \chi \ldots)$, or in $\mathcal{B}$

Identity introduction

$$\begin{array}{|ll} c = c & =\text{I} \end{array}$$

Identity elimination

$$\begin{array}{r|ll} m & a = b & \\ n & \mathcal{A}(\ldots a \ldots a \ldots) & \\ & \mathcal{A}(\ldots b \ldots a \ldots) & =\text{E } m,\ n \end{array}$$

$$\begin{array}{r|ll} m & a = b & \\ n & \mathcal{A}(\ldots b \ldots b \ldots) & \\ & \mathcal{A}(\ldots a \ldots b \ldots) & =\text{E } m,\ n \end{array}$$

C.7 Derived rules for FOL

$$\begin{array}{r|ll} m & \forall \chi \neg \mathcal{A} & \\ & \neg \exists \chi \mathcal{A} & \text{CQ } m \end{array}$$

$$\begin{array}{r|ll} m & \exists \chi \neg \mathcal{A} & \\ & \neg \forall \chi \mathcal{A} & \text{CQ } m \end{array}$$

$$\begin{array}{r|ll} m & \neg \exists \chi \mathcal{A} & \\ & \forall \chi \neg \mathcal{A} & \text{CQ } m \end{array}$$

$$\begin{array}{r|ll} m & \neg \forall \chi \mathcal{A} & \\ & \exists \chi \neg \mathcal{A} & \text{CQ } m \end{array}$$

In the Introduction to his volume *Symbolic Logic*, Charles Lutwidge Dodson advised: "When you come to any passage you don't understand, *read it again*: if you *still* don't understand it, *read it again*: if you fail, even after *three* readings, very likely your brain is getting a little tired. In that case, put the book away, and take to other occupations, and next day, when you come to it fresh, you will very likely find that it is *quite* easy."

The same might be said for this volume, although readers are forgiven if they take a break for snacks after *two* readings.

P.D. Magnus is an associate professor of philosophy in Albany, New York. His primary research is in the philosophy of science.

Tim Button is Senior Lecturer at University College London. His first book, *The Limits of Realism*, was published by Oxford University Press in 2013. His second book, *Philosophy and Model Theory*, was co-authored with Sean Walsh, and published by Oxford University Press in 2018.

Robert Trueman is Lecturer in Philosophy at the University of York. His first book, *Properties and Propositions*, was published by Cambridge University Press in 2021.