`forall`x ADELAIDE

forall$x$
Adelaide

Antony Eagle
University of Adelaide

Tim Button
University College London

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

© 2005–2021 by Antony Eagle, Tim Button and P.D. Magnus. Some rights reserved.

This version of `forall`x ADELAIDE is current as of 10 December 2021.

This book is a derivative work created by Antony Eagle, based upon Tim Button's 2016 Cambridge version of P.D. Magnus's `forall`x (version 1.29). There are pervasive substantive changes in content, theoretical approach, coverage, and appearance. (For one thing, it's more than twice as long.)

You can find the most up to date version of `forall`x ADELAIDE at `github.com/antonyeagle/forallx-adl`.

The current version of `forall`x Cambridge is available at `github.com/OpenLogicProject/forallx-cam`, which has now also diverged noticeably from the 2016 version. Magnus' original is available at `fecundity.com/logic`.

This book, like its predecessors, is released under a Creative Commons license (Attribution 4.0).

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit `creativecommons.org/licenses/by/4.0/` or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Typesetting was carried out entirely in XƎLaTeX using the `Memoir` class. The body text is set in Constantia; math is set in Cambria Math; sans serif in Calibri; and monospaced text in Consolas. The style for typesetting proofs is based on `fitch.sty` (v0.4) by Peter Selinger, University of Ottawa.

Contents

1 Key Notions 1

1 Arguments 2
2 Valid Arguments 5
3 Other Logical Notions 17

2 The Language of Sentential Logic 24

4 First Steps to Symbolisation 25
5 Connectives 36
6 Sentences of Sentential 50
7 Use and Mention 57

3 Truth Tables 63

8 Truth-Functional Connectives 64
9 Complete Truth Tables 75
10 Semantic Concepts 80
11 Entailment and Validity 85
12 Truth Table Shortcuts 93
13 Partial Truth Tables 98
14 Expressiveness of Sentential 105

4 The Language of Quantified Logic 110

15 Building Blocks of Quantifier 111

16 Sentences with One Quantifier 123
17 Multiple Generality 137
18 Identity 151
19 Definite Descriptions 160
20 Sentences of Quantifier 171

5 Interpretations 177

21 Extensionality 178
22 Truth in Quantifier 201
23 Semantic Concepts 217
24 Demonstrating Consistency and Invalidity 223
25 Reasoning about All Interpretations 229

6 Natural Deduction for Sentential 234

26 Proof and Reasoning 235
27 The Idea of a Formal Proof: Reasoning from Assumptions 239
28 Basic Rules for Sentential: Rules without Subproofs 244
29 Basic Rules for Sentential: Rules with Subproofs 255
30 Some Philosophical Issues about Conditionals and Meaning 280
31 Proof-Theoretic Concepts 284
32 Proof Strategies 287
33 Derived Rules for Sentential 289
34 Alternative Proof Systems for Sentential 298

7 Natural Deduction for Quantifier 302

35 Basic Rules for Quantifier 303
36 Derived Rules for Quantifier 323
37 Rules for Identity 328
38 Proof-Theoretic Concepts and Semantic Concepts 334
39 Next Steps in Logic 338

Appendices 340

A Alternative Terminology and Notation 340
B Quick Reference 346
C Index of defined terms 351

Acknowledgements, etc. 354

List of Figures

1 How the sections depend on one another. viii

4.1 A phrase structure tree for Example 18. 27
4.2 A phrase structure tree for Example 19. 27
4.3 Paraphrase of Example 26 showing its subsentential structure, as in Example 27. 30
4.4 Different sentential structures for 'Not A and B' shown in schematic syntactic trees. 30

6.1 Formation tree for '$\neg(P \wedge \neg(\neg Q \vee R))$'. 54

21.1 An interpretation represented diagrammatically. 188
21.2 A representation of 'Every Rose Has Its Thorn'. 189
21.3 Representing the argument from §15. 190
21.4 'Chat Systems', `https://xkcd.com/1810/` 190
21.5 A simple graph. 191
21.6 A graph with 'loops' 191
21.7 A more complicated graph. 192
21.8 Multiple techniques used to depict a complex interpretation. 192
21.9 A graph of a reflexive relation on $\{1, 3, 4\}$. 193
21.10 A graph of the transitive relation 'older than' on some University of Adelaide buildings. 195
21.11 An extract of Qantas' route network. 195
21.12 'next to': a symmetric but intransitive relation. 196
21.13 $>$ (black arrows) and $\geq$ (orange dotted arrows) on the domain $\{0, 1, 2\}$. 197
21.14 Black arrows indicate both $\subset$ and $\subseteq$, dotted arrows indicate $\subseteq$ only, on the domain $\wp\{a, b\} = \{\{a, b\}, \{a\}, \{b\}, \emptyset\}$. 198

29.1 Proof of $\neg(A \vee B) \therefore (\neg A \wedge \neg B)$ 275
29.2 Proof that $(A \vee B), \neg(A \wedge C), \neg(B \wedge \neg D) \therefore (\neg C \vee D)$. 276
29.3 A complicated proof 277

34.1 Schematic ∨E proof. 300
34.2 Schematic proof using DS to emulate ∨E. 300

How to Use This Book

This book has been designed for use in conjunction with the University of Adelaide courses PHIL 1110 *Introduction to Logic* and PHIL 1110L *Introductory Logic*. But it is suitable for self-study as well. I have included a number of features to assist your learning.

- `forall`x is divided into seven chapters, each further divided into sections and subsections. The sections are continuously numbered.
  - Chapter 1 gives an overview of how I understand the project of formal logic;
  - Chapters 2–3 cover sentential or truth-functional logic;
  - Chapters 4–5 cover quantified or predicate logic;
  - and Chapters 6–7 cover the formal proof systems for our logical languages.
- The book contains many cross-references to other sections. So a reference to '§6.2' indicates that you should consult section 6, subsection 2 – you will find this on page 50. Cross-references are hyperlinked, as are entries in the table of contents.
- Figure 1 shows how the sections depend on one another. For example, the arrows coming from §21 in the diagram show that understanding that section requires familiarity with §11 and §20, and also any sections on which they depend.
- Logical ideas and notation are pretty ubiquitous in philosophy, and there are a lot of different systems. We cannot cover all the alternatives, but some indication of other terminology and notation is contained in Appendix A.
- A quick reference to many of the aspects of the logical systems I introduce can be found in Appendix B.
- When I first use a new piece of technical terminology, it is introduced by writing it in small caps, LIKE THIS. You can find an index of defined terms in Appendix C.

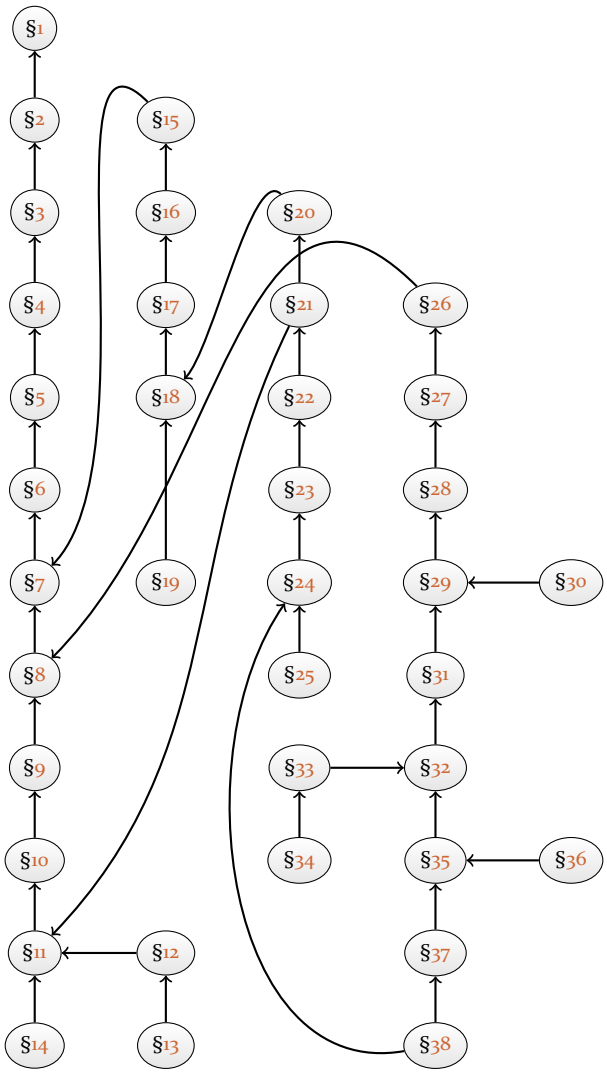


Figure 1: How the sections depend on one another.

- Each chapter in the book concludes with a box labelled 'Key Ideas in §*n*'. These are not a summary of the chapter, but contain some indication of what I regard as the main ideas that you should be taking away from your reading of the chapter.
- The book is the product of a number of authors. 'I' doesn't always mean me, but 'you' mostly means you, the reader, and 'we' mostly means you and me.

I appreciate any comments or corrections: `antony.eagle@adelaide.edu.au`.

Chapter 1

Key Notions

1

Arguments

Logic is the business of evaluating arguments – identifying some of the good ones and explaining why they are good. So what is an argument?

In everyday language, we sometimes use the word 'argument' to talk about belligerent shouting matches. Logic is not concerned with such teeth-gnashing and hair-pulling. They are not arguments, in our sense; they are disputes. A dispute like this is often more about expressing feelings than it is about persuasion.

Offering an ARGUMENT, in the sense relevant to logic (and other disciplines, like law and philosophy), is something more like *making a case*. It involves presenting reasons that are intended to favour, or support, a specific claim. Consider this example of an argument that someone might give:

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

We here have 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 the argument is well-constructed, the premises provide reasons in favour of the conclusion. In this example, the premises do seem to support the conclusion. At least they do, given the tacit assumption that you do not wish to get soaked.

So this is the sort of thing that logicians are interested in when they look at arguments. We shall say that an argument is any collection of premises, together with a conclusion.[1]

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.

[1] Because arguments are made of sentences, logicians are very concerned with the details of particular words and phrases appearing in sentences. Logic thus also has close connections with linguistics, particularly that sub-discipline of linguistics known as SEMANTICS, the theory of meaning.

Many arguments are expressed in this way. But a single sentence can contain a complete argument. Consider:

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

This argument has one premise followed by a conclusion.

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

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

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

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

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

so, therefore, hence, thus, accordingly, consequently, must

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.

In a good argument, the premises provide reasons for the conclusion. We are interested in understanding and analysing arguments because of this. Offering a good argument to someone emphasises some reasons that favour the conclusion. Reasonable people, we hope, may change their mind when they are given good reasons to do so. And so offering a good argument might *persuade* a reasonable person to accept its conclusion. We return to this issue of reasoning and argument a little later, in §2.3.

Above we said that when someone offers an argument, they intend to offer premises that support a given conclusion. But there is always a question as to whether some premises *really do* support a conclusion. Someone can offer a bad argument without knowing it, and thus intend to support some conclusion without managing to do so. Logicians aren't very interested in the intentions of people who might offer arguments, but they are very interested in whether the premises of a given argument do in fact support the conclusion. Some even think the central defining question of logic is the issue of how to characterise when a claim is a LOGICAL CONSEQUENCE of some other claims.

Key Ideas in §1

- An argument is a collection of sentences, divided into one or more premises and a single conclusion.
- The conclusion may be indicated by 'so', 'therefore' or other expressions; the premises indicated by 'since' or 'because'.
- The premises are intended to support the conclusion – though whether they do so is another matter.

Practice exercises

At the end of every section, there are practice exercises that review and explore the material covered in the chapter. There is no substitute for actually working through some problems, because logic is more about cultivating a *way of thinking* than it is about memorising facts.

A. What is the difference between argument in the everyday sense, and in the logicians' sense? What is the point of logical arguments?

B. 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.

2

Valid Arguments

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

There is a bassoon-playing dragon in Rundle Mall.
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, there are two sources of weakness. First: the argument's (only) premise is obviously false. Rundle Mall has some interesting buskers, but not quite *that* interesting. Second: the conclusion does not follow from the premise of the argument. Even if there were a bassoon-playing dragon in Rundle Mall, we would not be able to draw any conclusion about Dali's predilection for poker.

What about the main argument discussed in §1? The premises of this argument might well be false. It might be sunny outside; or it might be that you can avoid getting soaked without taking an umbrella. But even if both premises were true, it does not necessarily show you that you should take an umbrella. Perhaps you enjoy walking in the rain, and you would like to get soaked. So, even if both premises were true, the conclusion might nonetheless be false. (If we were to add the formerly tacit assumption that you do not wish to get soaked as a further premise, then the premises taken together would provide support for the conclusion.) Those premises might still provide some reason for thinking the conclusion is correct. But if the premises can be true, while the conclusion is false, there is at least some good sense in which there is a gap or lack in the support those premises provide for the conclusion.

The general point is as follows. For any argument, there are two ways that it might go wrong:

1. One or more of the premises might be false.
2. The conclusion might not follow from, or be a consequence of, the premises – even if the premises were true, they would not support the conclusion.

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

2.2 Conclusive Arguments

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

> An argument is CONCLUSIVE if, and only if, the truth of the premises guarantees the truth of the conclusion.
> In other words: an argument is conclusive if, and only if: it is not possible for the premises of the argument to be true while the conclusion is false.

Consider another argument:

You are reading this book.
This is a logic book.
So: You are a logic student.

This is not a terrible argument. Both of the premises are true. And most people who read this book are logic students. Yet, it is possible for someone besides a logic student to read this book. If your housemate picked up the book and thumbed through it, they would not immediately become a logic student. So the premises of this argument, even though they are true, do not guarantee the truth of the conclusion. This is not a conclusive argument.

Or consider this pair of arguments:

Mary has two brothers; So: There are three siblings in her family.	Mary has two brothers; So: There are *at least* three siblings in her family.

The argument on the left might be pretty good: that premise provides some reason to accept the conclusion, especially if you imagine a real conversation in which someone makes this case. It would be weird to use a premise about Mary's brothers in an argument for a conclusion about her siblings unless Mary had no sisters. Weird, but

not impossible. Strictly speaking, if Mary had a sister the speaker did not mention, it would be possible for the premise to be true while the conclusion is false. So the left argument is not conclusive, interpreted strictly.

The argument on the right shows that the premise does make a compelling case for a related but more hedged conclusion. You might think 'that second argument is a bit nit-picking'. But that is exactly what makes it so watertight. It doesn't depend on what would make for a 'normal' conversation, or whether you are making the same assumptions as the speaker, etc. No matter what, the truth of the premises secures the truth of the conclusion: it is conclusive.

The crucial thing about a conclusive argument is that it is impossible, in a very strict sense, for the premises to be true whilst the conclusion is false. Consider this example:

Oranges are either fruits or musical instruments.
Oranges are not fruits.
So: Oranges are musical instruments.

The conclusion of this argument is ridiculous. Nevertheless, it follows from the premises. *If* both premises were true, *then* the conclusion would just have to be true. So the argument is conclusive.

Why is this argument conclusive? The most important factor for us in considering what makes an argument conclusive is to examine the argument's STRUCTURE – the grammatical forms of the premises and conclusion. An argument will be conclusive if its structure *guarantees* that its premises *support* the conclusion. In the present case, one premise says that oranges are in one of two categories; the other premise says that oranges are not in the first category. We conclude that they are in the second category. The premises and conclusion are about oranges. But it is plausible to think that *any* argument with this same sort of structure must be conclusive, whether we are talking about oranges, or cars – or anything really.

Conclusiveness of an argument is insensitive to the truth or falsity of the premises. An argument can be conclusive while nevertheless going wrong in the first of the ways we identified in §2.1. Consider this argument

The earth has twenty-eight moons;
So: The earth has an even number of moons.

The argument is conclusive, as there is no possible way in which the earth could have twenty-eight moons while having an odd number of moons. But the premise is so obviously false that this argument could never be used to persuade anyone. The premise supports the conclusion, but that support is moot given the evident falsity of the premise.

2.3 Reasons to Believe

A conclusive argument, in the logician's sense, links the premises to the conclusion. It turns the reasons you have for accepting to the premises into reasons to accept its

conclusion. But a conclusive argument need not provide you with a reason to believe the conclusion. One way this can happen is when you don't accept any of the premises in the first place. When the premises support the conclusion, that might just mean that they *would* be excellent reasons to accept the conclusion – *if only they were true*!

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

But even this may be doubted. Often, when you believe the premises, a conclusive argument provides you with a reason to believe the conclusion. In that case, it might be appropriate for you to infer the conclusion from the premises.

But sometimes a conclusive argument shows that some premises support a conclusion you cannot accept. Suppose, for example, that you know the conclusion to be false. The fact that the argument is conclusive and has a false conclusion tells you that the premises cannot all be true. (Consider the argument from the previous section with the false conclusion 'Oranges are musical instruments': the second premise is as absurd as the conclusion.) In general, when an argument is conclusive

- the truth of all the premises guarantees the truth of the conclusion; and equally
- the falsity of the conclusion guarantees the falsity of at least one of the premises.

In this sort of situation, you might find that the argument gives you a better reason to abandon any belief in one of the premises than to accept the conclusion. A conclusive argument shows there is *some* reason to believe its conclusion, if you accept its premises; it doesn't mean there aren't *better* reasons to reject its premises, if you reject its conclusion. Consider this sort of example:[1]

If I look in the cupboard, I'll find some muesli.
I am looking in the cupboard.
So: I find some muesli.

Someone might believe both premises, and not accept the conclusion, because on looking in the cupboard, they do not see the muesli. (Someone else finished it off earlier.) It would be silly for this person to 'follow the argument where it leads'. Rather, they should use the fact that these premises entail a conclusion they now know to be false, having just looked, to reject one of the premises. The obvious candidate is the first premise. So this person should probably stop believing that they'll find muesli if they look in the cupboard, and start believing instead that they are out of muesli or suchlike.

Some cases are less straightforward. Consider this argument:

It is immoral to cause avoidable suffering.
Eating meat causes avoidable suffering.
So: Eating meat is immoral.

[1] This sort of example is discussed by Gilbert Harman, *Change in View*, MIT Press, esp. ch. 2.

Many people will find the premises plausible, and the conclusion therefore compelling. This is part of the case for vegetarianism. But not everyone finds the conclusion acceptable; such people end up finding one or both of the premises should be rejected. Interestingly, people may find themselves still finding the premises attractive even when they recognise a conclusion follows that they cannot accept. Here they may find that there is some reason to accept the premises (perhaps they seem true at first glance), and some reason to reject them (they have, at second glance, consequences the person cannot accept). I don't want to adjudicate the merits of this argument here. I only want to emphasise that even offering a conclusive argument to someone, with premises that they currently accept, is not enough to make them come to believe the conclusion.

Sometimes people think that logic will provide a powerful tool to persuade and convince others of their point of view. They sometimes want to study logic as if it is some dark art enabling them to subdue beliefs that contradict their own. This is not really a very nice thing to want to do – to force a belief on someone, whether they want to believe it or not – and so it is not particularly regrettable that logic doesn't help you to do it. Logic can show which claims follow from which others, and which contradict one another. It can help us elaborate the content of some claim, or delineate the commitments a certain belief would incur. But logic does not tell you what to believe, even when you have a conclusive argument.

The question, *what ought I to believe?* is one of the deepest in the area of philosophy known as EPISTEMOLOGY, the theory of knowledge. Logic is not able to answer that question all by itself. Even if logic tells you that there is a conclusive argument from premise $\mathcal{A}$ to conclusion $\mathcal{B}$, logic can't tell you whether you ought to believe both, or reject both. However, logic will tell you something important, even if it is only a limited part of the answer to the question of rational belief. It will tell you that, when you know an argument to be conclusive, *you cannot both accept its premises while rejecting its conclusion*. Thus conceived, logic is not even a science of reasoning, because it does not tell you what to think. Logic can't tell you, or anyone else, which packages of premises-and-conclusions to accept.

2.4 Conclusiveness for Special Reasons

An argument can be conclusive for reasons unrelated to its structure. Take this example:

Juanita is a vixen.
So: Juanita is a fox.

It is impossible for the premise to be true and the conclusion false. So the argument is conclusive. But this is not due to the structure of the argument. Here is an inconclusive argument with seemingly the same structure or form. The new argument is the result of replacing the word 'vixen' in the first argument with the word 'cathedral', but keeping the overall grammatical structure the same:

Juanita is a cathedral.
So: Juanita is a fox.

This might suggest that the conclusiveness of the first argument *is* keyed to the meaning of the words 'vixen' and 'fox'. But, whether or not that is right, it is not simply the FORM of the argument that makes it conclusive. It is instructive to compare the first argument with this modification, where we replace 'vixen' with the near-synonym 'female fox':

Juanita is a female fox.
So: Juanita is a fox.

This also seems to be conclusive. But now we might suspect the occurrence of the word 'fox' in both premise and conclusion is not mere coincidence, but an essential part of the explanation as to why this is conclusive.

Equally, consider the argument:

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

Again, because nothing can be both green all over and red all over, the truth of the premise would guarantee the truth of the conclusion. So the argument is conclusive. But here is an inconclusive argument with the same form:

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

The argument is inconclusive, since it is possible to be green all over and shiny all over. (I might paint my nails with an elegant shiny green varnish.) Plausibly, the conclusiveness of this argument is keyed to the way that colours (or colour-words) interact. But, whether or not that is right, it is not simply the form of the argument that makes it conclusive.

An argument can be conclusive due to its structure, and also be conclusive for other reasons. Arguably, this might be going on in the argument discussed at the end of §2.2, with the premise 'Oranges are not fruits'. Some people might think this premise has to be false, because of what oranges are. (Many will say that *being a fruit* is an essential part of what it is to be an orange.) But if the premise 'Oranges are not fruits' has to be false, it is not possible for the premises to be true. So it is not possible for premises to be true *while* the conclusion is false. Hence the argument is conclusive – both because it has a good structure, but also because it has a premise that cannot be true.[2]

[2] When an argument has an impossible premise, any argument with that premise will be conclusive *no matter what* the conclusion is! So this is a weird kind of case of conclusiveness. But nothing much really turns on it, and it is simpler to simply count it as conclusive than to try and separate out such 'degenerate' cases of conclusive arguments. See also §3.3.

2.5 Validity

Logicians try to steer clear of controversial matters like whether there is a definition of an orange that requires it to be a fruit, or whether there is a 'connection in meaning' between being green and not being red. It is often difficult to figure such things out from the armchair (a logician's preferred habitat), and there may be widespread disagreement even among subject matter experts.

So logicians do not study conclusive arguments in general, but rather concentrate on those conclusive arguments which have a good structure or form.[3] This is why the logic we are studying is sometimes called FORMAL LOGIC. We introduce a special term for the class of arguments logicians are especially interested in:

> An argument is VALID if, and only if, it is conclusive due to its structure; otherwise it is INVALID.

The notion of the structure of a sentence, or an argument, is an intuitive one. I make the notion more precise in §4.1. Relying on our intuitive grasp of the notion for now, however, we can see the argument about ogres on the right has the same form as the argument on the left about oranges (slightly tweaked from our earlier presentation in §2.2 to make its structure clearer). It is easy to see that both of these arguments are conclusive and valid:

	Either Oranges are fruits or oranges are musical instruments.		Either Ogres are fearsome or ogres are mythical.
	It is not the case that Oranges are fruits.		It is not the case that Ogres are fearsome.
So:	Oranges are musical instruments.	So:	Ogres are mythical.

The shared structure of these two arguments is something like this:

Either $\mathcal{A}$ or $\mathcal{B}$.
It is not the case that $\mathcal{A}$.
So: $\mathcal{B}$.

Any argument with this structure will be conclusive in virtue of structure, and hence valid. It does not matter, really, what sentences we put in place of '$\mathcal{A}$' and '$\mathcal{B}$'. (Within limits: you can't put a question or an exclamation and get a valid argument – see §3.1.)

This highlights that valid arguments do not need to have true premises or even true conclusions. We can put a true sentence in place of $\mathcal{A}$ and a false sentence in place of $\mathcal{B}$, and both premises and the conclusion will be false. The argument is still valid.

[3] It can be very hard to tell whether an invalid argument is conclusive or inconclusive. Consider the argument 'The sea is full of water; so the sea is full of H_2O'. This is conclusive, since water just is the same stuff as H_2O. The sea cannot be full of that water stuff without being full of that exact same stuff, namely, H_2O stuff. But it took a lot of chemistry and ingenious experiments to figure out that water is H_2O. So it was not at all obvious that this argument was conclusive. On the other hand, it is generally very clear when an argument is conclusive due to its structure – you can just see the structure when the argument is presented to you.

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 inconclusive, and hence invalid.

Return briefly to another example we discussed earlier:

Juanita is a female fox.
So: Juanita is a fox.

This seems to have something like this structure:

a is an $\mathcal{F}$ $\mathcal{G}$.
So: a is a $\mathcal{G}$.

For most adjectives $\mathcal{F}$, this structure yields a conclusive argument when you replace the schematic letters by English words. E.g.,

Bob is a tall man;
So: Bob is tall.

But not all: some adjectives like 'fake' or 'alleged' do not yield conclusive arguments when substituted for $\mathcal{F}$: 'This is a fake gun; so this is a gun' is not a conclusive argument. We will return in §15 to the logical structure of examples like these, and to expressions like 'fake gun' in §16.6.

2.6 Soundness

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 the structure of the argument ensures that the premises support the conclusion. Nonetheless, we shall say that an argument is SOUND if, and only if, it is both valid and all of its premises are true. So every sound argument is valid and conclusive. But not every valid argument is sound, and not every conclusive argument is sound.

It is often possible to see that an argument is valid even when one has no idea whether it is sound. Consider this extreme example (after Lewis Carroll's *Jabberwocky*):

'Twas brillig, and the slithy toves did gyre and gimble in the wabe.
So: The slithy toves did gyre and gimble in the wabe.

This argument is valid, simply because of its structure (it has a premise conjoining two claims by 'and', and a conclusion which is one of those claims). But is it sound? That would depend on figuring out what all those nonsense words mean!

2.7 Inductive Assessment of Arguments

Many good arguments are inconclusive and invalid. Consider this one:

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

This argument generalises from observations about several cases to a conclusion about all cases. Though it is invalid, that doesn't mean it is a bad argument. The premises appear to provide some support for the conclusion, though it falls short of being conclusive.

INDUCTION describes a form of reasoning from evidence to hypotheses about that evidence. For example, when we reason from a sample of eligible voters to a hypothesis about how the whole population will vote, we are reasoning inductively. INDUCTIVE LOGIC is the attempt to generalise deductive logic to evaluate arguments in line with the canons of good inductive reasoning. The proponents of inductive logic think there might be a generalisation of deductive logic which enables us to evaluate arguments in a more fine-grained way than the options we've canvassed so far (i.e., just 'valid', and 'invalid and conclusive' and 'invalid and inconclusive'). A significant part of the project of inductive logic is the attempt to classify inconclusive arguments in terms of whether their premises provide good inductive evidence in favour of their conclusions.

In the example, the premises are of the form 'In January n, it rains', and the conclusion is of the form 'Every January, it rains'. This argument thus has a general conclusion drawn from instances of that generalisation. (That is, 'it rains in January 1997' is an instance of the generalisation 'it rains in January of every year'.) If we regard the foregoing premises as providing decent support for the conclusion, we might think that adding additional premises of the same sort before drawing the conclusion would make it even stronger: In January 2001, it rained in London; In January 2002.... This principle, that a generalisation is increasingly supported the more instances we adduce, more might be part of our toolkit when evaluating arguments inductively. But, no matter how many premises of this form we add, the argument will remain inconclusive. Even if it has rained in London in every January thus far, it remains *possible* that London will stay dry next January. (Even if we include every instance – past, present, and future – the argument will be inconclusive, because one will need the additional premise *that there are no other instances than those we've included*.)

The point of all this is that most arguments which are inductively very strong are not (deductively) valid. Arguments which represent very good examples of inductive reasoning generally 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, our interest is simply in sorting the (deductively) valid arguments from the invalid ones. The project of inductive logic, of sorting the invalid arguments further into the inductively good and poor ones, we shall set aside entirely from here on.

2.8 Making Conclusive Arguments Valid

Some arguments which are conclusive but invalid can be turned into valid arguments. So consider again the argument 'The sculpture is green all over; therefore it is not red all over'. We can make a valid argument from this by adding a premise:

The sculpture is green all over.
If the sculpture is green all over, then it is not red all over.
So: The sculpture is not red all over.

This new argument has a premise which makes explicit a fact about green and red that was merely implicit in the original argument. Since the original argument was conclusive – since the fact about green and red is true just in virtue of the meaning of the words 'green' and 'red' (and 'not') – the new argument remains conclusive. (We can't undermine conclusiveness by adding further premises.) But the new argument is valid, because the additional premise we have added yields an argument with a structure that guarantees the truth of the conclusion, given the truth of the premises.

The original argument is sometimes thought to be merely an abbreviation of the expanded valid argument. An argument with an unstated premise, such that it can be seen to be valid when the premise is made explicit, is called an ENTHYMEME.[4] Many inconclusive arguments can be treated as enthymematic, if the unstated premise is obvious enough:

The Nasty party platform includes imprisoning people for chewing gum;
So: The Nasty party will not form the next government.

The unstated premise is something like 'If a party platform includes imprisoning people for chewing gum, then that party will win too few votes to form the next government'. The unstated premise may or may not be true. But if it is added, the argument is made valid.

Any conclusive argument you are likely to come across will either already be valid, or can be transformed into a valid argument by making some assumption on which it implicitly relies into an explicit premise.

Not every inconclusive argument should be treated as an enthymeme. In particular, many strong inductive arguments can be made weaker when they are treated as enthymematic. Consider:

[4] The term is from ancient Greek; the concept was given its first philosophical treatment by Aristotle in his *Rhetoric*. He gives this example, among others: 'He is ill, since he has fever'.

In January 2015, it was hot in Adelaide.
In January 2020, it was hot in Adelaide.
So: In January 2025, it will be hot in Adelaide.

This argument is inconclusive. It can be made valid by adding the unstated premise 'Every January, it is hot in Adelaide'. But that unstated premise is extremely strong – we do not have sufficient evidence to conclude that it will be hot in Adelaide in January *for eternity*. So while the premises we have been given explicitly are good reason to think Adelaide will continue to have a hot January for the foreseeable future, we do not have good enough reason to think that every January will be hot. Treating the argument as an enthymeme makes it valid, but also makes it less persuasive, since the unstated premise on which it relies is not one many people will share.[5]

Key Ideas in §2

- An argument is conclusive if, and only if, the truth of the premises guarantees the truth of the conclusion.
- An argument is valid if, and only if, the form of the premises and conclusion alone ensures that it is conclusive. Not every conclusive argument is valid (though they can be made valid by addition of appropriate premises).
- An argument can be good and persuade us of its conclusion even if it is not conclusive; and we can fail to be persuaded of the conclusion of a conclusive argument, since one might come to reject its premises.

Practice exercises

A. What is a *conclusive* argument? What, in addition to being conclusive, is required for an argument to be *valid*? What, in addition to being valid, is required for an argument to be *sound*?

B. Which of the following arguments are valid? Which are invalid but conclusive? Which are inconclusive? Comment on any difficulties or points of interest.

1. 1. Socrates is a man.
   2. All men are carrots.
   So: Therefore, Socrates is a carrot.

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

5 If we had claim that the unstated premise was rather 'If it has been hot in January in recent representative years, it will be hot in January for the near future', we would have had a valid argument and one that has a plausible unstated premise – though the unstated premise seems to be plausible only because the unamended inductive argument was fine to begin with!

3.
1. Abe Lincoln was the president of the United States.
So: Abe Lincoln was a citizen of the United States.

4.
1. If I pull the trigger, Abe Lincoln will die.
2. I do not pull the trigger.
So: Abe Lincoln will not die.

5.
1. Abe Lincoln was either from France or from Luxembourg.
2. Abe Lincoln was not from Luxembourg.
So: Abe Lincoln was from France.

6.
1. If the world ends today, then I will not need to get up tomorrow morning.
2. I will need to get up tomorrow morning.
So: The world will not end today.

7.
1. Joe is right now 19 years old.
2. Joe (the same one) is also right now 87 years old.
So: Bob is now 20 years old.

C. Which of the following are valid arguments?

1.
Lizards are cute;
So: lizards are cute, or I'll eat my hat.

2.
Anyone can pass logic if they work diligently.
Sarah is diligent and hard working.
So: Sarah can pass logic.

3.
If you buy a lottery ticket, you've wasted your money.
You have wasted your money.
So: you have bought a lottery ticket

4.
We must do something.
Quacking like a duck is something.
So: We must quack like a duck.

D. 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.

3

Other Logical Notions

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, where the premises are supposed to support the conclusion. But if premises are supposed to state reasons, and conclusions are supposed to state claims, then both of them have to be the sort of sentence which can be used to *say how things are*, truly or falsely.

So 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!'
- **Cohortatives**, e.g., 'Let's go to the beach!'
- **Exclamations**, e.g., 'Ouch!'

The common feature of these three kinds of sentence is that they cannot be used to make *assertions*: 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 sentences that are capable of having a TRUTH VALUE. The notions of validity and conclusiveness are defined in terms of truth preservation, so these properties depend on the constituents of arguments being the kinds of things which have a truth value.

> A DECLARATIVE SENTENCES (like 'Bob is singing') states how things are or could be; it is a sentence that is either true or false.

The two truth values that concern us are just TRUE and FALSE. We may not know which truth value a declarative sentence has, but we generally know what kinds of conditions would need to obtain in order for it to be true or false. (We thus have no need of a supposed intermediate truth value like 'unknown' – that would reflect our attitudes or beliefs about a claim, whereas the truth value reflects how things really are independently of our attitudes or beliefs.)

To form part of an argument, a sentence must have the kind of grammatical structure that permits it to have a truth value. In terms of the notion of structure introduced in §2.2, we are going to focus on those structures which yield a declarative sentence when supplied with declarative sentences. For example 'it is not the case that $\mathcal{A}$' yields a declarative sentence whenever we put a declarative sentence in for $\mathcal{A}$, and may not yield anything grammatical at all otherwise:

1. It is not the case that dogs can fly. (Declarative, true)
2. It is not the case that dogs bark. (Declarative, false)
3. It is not the case that in 10,000BC, Kaurna people occupied Kangaroo Island. (Declarative, unknown but determinate truth value)
4. It is not the case that are you feeling sleepy? (Ungrammatical)

3.2 Consistency

Consider these two sentences:

5. Jane's only brother is shorter than her.
6. Jane's only brother is taller than her.

Logic alone cannot tell us which, if either, of these sentences is true. Yet we can say that *if* the first sentence 5 is true, *then* the second sentence 6 must be false. And if 6 is true, then 5 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.

Conversely, 5 and 6 are JOINTLY INCONSISTENT.

Consistency is relatively trivial in some cases. If we take at random some unrelated sentences, they will typically be consistent. For example, 'Eggs are delicious', 'Frogs hop', and 'Barry hosts a podcast' have nothing much to do with one another. It is unsurprising to find out that they can all be true together, because how could unrelated sentences place any constraints on the possibility of their simultaneous truth? But consistency can be surprising, when related sentences can be true together even though it may at first glance seem impossible. One of Einstein's great contributions was the discovery that these claims are consistent: 'Albert correctly measured the spaceship to

be 50m', and 'Brunhilde correctly measured the spaceship to be 45m', so long as Albert and Brunhilde are in relative motion.

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

7. There are at least four giraffes at the wild animal park.
8. There are exactly seven gorillas at the wild animal park.
9. There are not more than two Martians at the wild animal park.
10. Every giraffe at the wild animal park is a Martian.

7 and 10 together entail that there are at least four Martian giraffes at the park. This conflicts with 9, which implies that there are no more than two Martian giraffes there. So the sentences 7–10 are jointly inconsistent. They cannot all be true together. (Note that the sentences 7, 9 and 10 are jointly inconsistent. But if some sentences are already jointly inconsistent, adding an extra sentence to the mix will not make them consistent!)

There is an interesting connection between consistency and conclusive arguments. A conclusive argument is one where the premises guarantee the truth of the conclusion. So it is an argument where if the premises are true, the conclusion must be true. So the premises cannot be jointly consistent with the claim that the conclusion is false. Since the argument 'Dogs and cats are animals, so dogs are animals' is conclusive, that shows that the sentences 'Dogs and cats are animals' and 'Dogs are **not** animals' are jointly inconsistent. If an argument is conclusive, the premises of the argument taken together with the denial of the conclusion will be jointly inconsistent.

> Sentences $\mathcal{A}$ and $\mathcal{B}$ are jointly inconsistent iff the arguments' $\mathcal{A} \therefore$ not-$\mathcal{B}$' and '$\mathcal{B} \therefore$ not-$\mathcal{A}$' are both conclusive. This can be generalised to arbitrary inconsistent collections of sentences.

We just linked consistency to conclusive arguments. There is an analogous notion linked to valid arguments:

> Sentences are JOINTLY FORMALLY CONSISTENT if, and only if, considering only their structure, they can all be true together.

Another way to put it: some sentences are formally consistent iff, looking just at their structure (and not looking at what they are actually about), they might all be true together.

Just as validity is more stringent than conclusiveness (it is conclusiveness plus something more), consistency is more stringent than formal consistency (it is formal consistency plus substantive consistency). If some sentences are jointly consistent, they are also jointly formally consistent. *But some formally consistent sentences are jointly inconsistent*. Any conclusive but invalid argument will give us an example.

For example, since 'The sculpture is green all over, so the sculpture is not red all over' is conclusive, these sentences are jointly formally consistent but not consistent:

11. The sculpture is uniformly green all over.
12. The sculpture is uniformly red all over.

These sentences have have no interesting internal structure for our purposes, so it is easy to make them formally consistent. But, holding fixed their actual meaning (particularly the actual meaning of 'red' and 'green'), we see that the truth of 11 excludes the truth of 12. (Again, more on the notion of structure invoked here in §4.1.)

3.3 Necessity and Contingency

In assessing whether an argument is conclusive, we care about what would be true *if* the premises were true. But some sentences just *must* be true. Consider these sentences:

13. It is raining.
14. If it is raining, water is precipitating from the sky.
15. If something is green, it is colourless.
16. Either it is raining here, or it is not.
17. It is both raining here and not raining here.

In order to know if sentence 13 is true, you would need to look outside or check the weather channel. It might be true; it might be false.

Sentence 14 is different. You do not need to look outside to know that it says something true. Regardless of what the weather is like, if it is raining, water is precipitating – that is just what rain *is*, meteorologically speaking. That is a NECESSARY TRUTH. Here, a necessary connection in meaning between 'rain' and 'precipitation' makes what the sentence says true in every circumstance.

Sentence 15 is a NECESSARY FALSEHOOD or IMPOSSIBILITY. Nothing is, or even could be, both green and colourless. We don't need to do any scientific or other investigation to know that 15 is not and cannot be true.

Sentence 16 is also a necessary truth. Unlike sentence 14, however, it is the structure of the sentence which makes it necessary. No matter what 'raining here' means, 'Either it is raining here or it is not raining here' will be true. The structure 'Either it is ... or it is not ...', where both gaps ('...') are filled by the same phrase, must be a true sentence.

Equally, you do not need to check the weather, or even the meaning of words, to determine whether or not sentence 17 is true. It must be false, simply as a matter of structure. It might be raining here and not raining across town; it might be raining now but stop raining even as you finish this sentence; but it is impossible for it to be both raining and not raining in the same place and at the same time. So, whatever the world is like, it is not both raining here and not raining here. It is a NECESSARY FALSEHOOD.

These last two examples, of necessary truths and impossibilities in virtue of structure, are of particular interest to logicians. We will come back to them in §10.

A sentence which is capable of being true or false, but which says something which is neither necessarily true nor necessarily false, is CONTINGENT.

If a sentence says something which is sometimes true and sometimes false, it will definitely be contingent. But something might *always* be true and still be contingent. For instance, it seems plausible that whenever there have been people, some of them habitually arrive late. 'Some people are habitually late' is always true. But it is contingent, it seems: human nature could have been more punctual. If so, the sentence would have been false. But if something is really necessary, it will always be true, and couldn't even possibly be false.[1]

If some sentences contain amongst themselves a necessary falsehood, those sentences are jointly inconsistent. At least one of them cannot be true, so they cannot all be true together. Accordingly, if an argument has a premise that is a necessary falsehood, or its conclusion is a necessary truth, or both, then the argument is conclusive – its premises and the denial of its conclusion will be jointly inconsistent.

- An argument with a necessarily true conclusion is conclusive;
- An argument with an impossible premise is conclusive.

This second observation might be surprising. But note that if a premise is impossible, there is no way to make it true, and hence no way to make it true *while* making the conclusion false. These are both *degenerate* cases of conclusiveness, where there need be no real connection between the premises and conclusion to ground the conclusiveness of an argument.

Key Ideas in §3

- Arguments are made up of declarative sentences, all of which are either true or false.
- Some declarative sentences are formally consistent if, and only if, their structures don't rule out the possibility that they are all true together.
- Some declarative sentences can only have one truth value – they are either necessary or impossible. Others are contingent, having one truth value in some circumstances and the other truth value in other circumstances.

[1] Here's an interesting example to consider. It seems that, whenever anyone says the sentence 'I am here now', they say something true. That sentence is, whenever it is uttered, *truly uttered*. But does it say something necessary or contingent?

Practice exercises

A. Which of the following sentences are capable of being true or false?

1. Earth is the third planet from the Sun.
2. Pluto is the ninth planet from the Sun.
3. Have you been feeding the lions?
4. Socrates said, 'Be as you wish to seem'.
5. 'Have you been feeding the lions?' is a sentence.
6. Always forgive your enemies; nothing annoys them so much.

B. Which of the following are declarative sentences?

1. Answer me!
2. 'Answer me!', she demanded.
3. You are required to answer me.
4. Saying nothing is not an answer.
5. If you want answers, ask Alfred.
6. Why won't you answer me?
7. All that is left to ask is: 'Who has the answers?'

C. 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.

D. Look back at the sentences 7–10 in this section (about giraffes, gorillas and Martians in the wild animal park), and consider each of the following:

1. 8, 9, and 10
2. 7, 9, and 10
3. 7, 8, and 10
4. 7, 8, and 9

Which are jointly consistent? Which are jointly inconsistent?

E. Are these sentences jointly consistent?

1. There are three people leaving the party: Atheer, Brigitte, and James.
2. Brigitte is wearing Atheer's hat.
3. Each of the people is wearing a hat.
4. No person is wearing their own hat.
5. Atheer is wearing Brigitte's hat.

F. Could there be:

1. A conclusive argument, the conclusion of which is necessarily false?
2. An inconclusive 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.

G. Some feature of a set *A* is MONOTONIC if any set including all the members of *A* also has the feature. (So *having at least 3 members* is monotonic, since adding more members clearly preserves that feature.) Explain why inconsistency of a set of sentences is monotonic.

Chapter 2

The Language of Sentential Logic

4

First Steps to Symbolisation

4.1 Argument Structure

Consider this argument:

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

and another argument:

> Jenny is an anarcho-syndicalist.
> If Jenny is an anarcho-syndicalist, then Dipan is an avid reader of Tolstoy.
> So: Dipan is an avid reader of Tolstoy.

Both arguments are valid, and there is a straightforward sense in which we can say that they share a common structure. We might express the structure thus, when we let letters stand for phrases in the original argument:

> A
> If A, then C
> So: C

This is an excellent argument STRUCTURE. Surely any argument with *this* structure will be valid. And this is not the only good argument structure. Consider an argument like:

> Jenny is either happy or sad.
> Jenny is not happy.
> So: Jenny is sad.

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

A or B
not: A
So: B

A superb structure! You will recall that this was the structure we saw in the original arguments which introduced the idea of validity in §2.5. And here is a final example:

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

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

not both: A and B
A
So: not: B

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

4.2 Sentence Trees and Canonical Clauses

Since arguments are made of sentences, the logical structure of an argument will be related to the grammatical structure of the sentences within it. We've already seen in §3.1 that arguments are made up of declarative sentences. One standard way of analysing the grammatical structure of a declarative sentence is to break it into constituent phrases. Such approaches are known as PHRASE STRUCTURE GRAMMARS. These structures are usefully depicted in a hierarchical SYNTACTIC TREE. Consider the example:

18. Tariq is a man.

This sentence (phrase of type S) can be divided into two main parts: the NOUN PHRASE 'Tariq' (type NP), and the VERB PHRASE 'is a man' (type VP). In this example, the verb phrase itself divides into the verb 'is' and the DETERMINER PHRASE 'a man' (type DP). We will return to the internal structure of sentences in §15. This phrase structure is depicted in the tree in Figure 4.1.

Let's look at a more complicated example. Consider

19. Alice will ace the test, or Bob will.

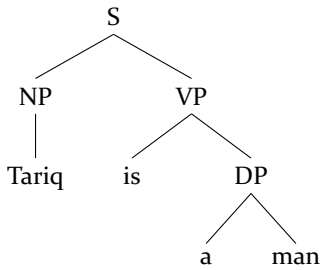


Figure 4.1: A phrase structure tree for Example 18.

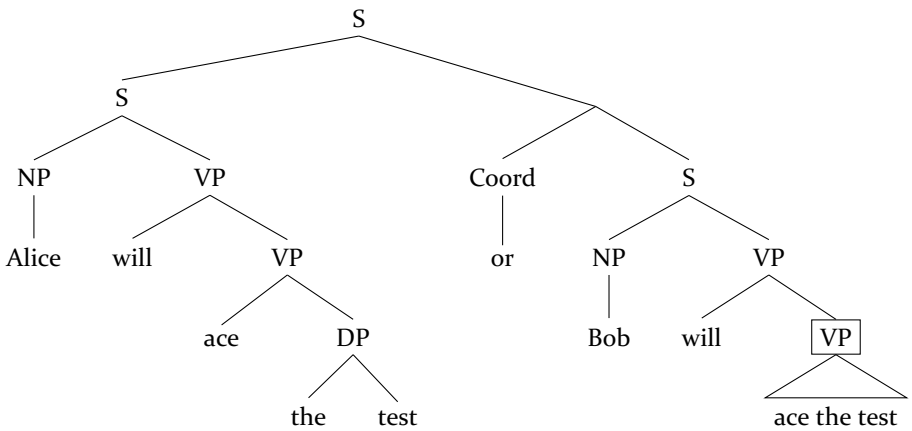


Figure 4.2: A phrase structure tree for Example 19.

First we note that this sentence is elliptical, the verb phrase 'ace the test' being omitted after 'Bob will' as it is understood to be supplied by the first clause of the sentence.[1] The full tree, with the elided phrase supplied (marked by having the VP label in a box at the lower right), is depicted in Figure 4.2. In this example, the whole sentence is a compound of two clauses which are sentences in their own right, or SUBSENTENCES, connected by 'or'.

When analysing the structure of compound sentences, a useful notion is that of a CANONICAL CLAUSE.[2] These are the simplest units of English sentences that can constitute a sentence by themselves. Here are some examples:

[1] A closely related example is discussed by Paul Elbourne (2011) *Meaning*, Oxford University Press, pp. 74–6. He argues that this kind of elision is crucial evidence for phrase structure grammars, for only phrases can be omitted in this way: witness the ungrammatical 'Alice will ace the test and Bob will the', where arbitrary words are omitted that do not form a grammatical phrase. This is evidence that English syntax is sensitive to the phrasal structure of sentences, and does not treat them as merely a string of words.

[2] A fuller treatment of canonical clauses can be found in Rodney Huddleston and Geoffrey Pullum (2005) *A Student's Introduction to English Grammar*, Cambridge University Press, pp. 24–5.

20. They knew the victim.
21. She has read your article.
22. Jenny is happy.

A canonical clause has internal structure too, but its parts are not themselves sentences. It is composed of a GRAMMATICAL SUBJECT, typically but not always a noun phrase ('Jenny', 'She'), and a PREDICATE, always a verb phrase ('knew the victim', 'is happy').

The following, in contrast with the previous examples, are noncanonical clauses:

23. They did not know the victim.
24. She has read your article and she vehemently disagrees with it.
25. She says that Jenny is happy.

These give us some characteristics of canonical clauses:

- They are *positive*, as in 20, rather than negative, as indicated by the underlined 'not' in 23.
- Canonical clauses are simple, and not coordinated with any other clause, as in 21. In 24, the COORDINATOR 'and' links two clauses that would be canonical on their own into a longer COMPOUND sentence.
- The underlined clause in 25 is a *subordinate* clause, part of a more complex clause. Canonical clauses are main clauses, as in 22.

4.3 Identifying Argument Structure

Our topic is not English grammar. But the idea of a canonical clause is illuminating for the logic we are examining in this part of the course. Sentential logic is, more or less, the logic that helps us understand and analyse arguments whose conclusiveness depends on their constituent canonical clauses and how they are joined together by SENTENCE CONNECTIVES. The sentence connectives we focus on are a grammatically diverse group, including

Coordinators including 'and', 'or', 'if and only if ' (also 'but')

Adjunct heads including 'if ... then ...', 'only if' (also 'unless', 'because', 'since', 'hence' – though we treat these last as signalling premises and conclusions of arguments)

Negatives including 'not', '-n't', 'it is not the case that' ('never').

And there are others that we won't deal with ('always', 'might').

How can we identify the logical structure of an argument? The logical structure of an argument is the form one arrives at by 'abstracting away' the details of the words in an argument, *except for a special set of words*, the STRUCTURAL WORDS. This connects with what we have just been saying, because in many of the examples from §4.1,

the arguments can be analysed as involving canonical clauses which are linked by the structural words 'and', 'or', 'not', 'if ... then ...' and '... if and only if ...'. So our efforts to identify the logical structure of arguments often coincide with linguist's efforts to identify the canonical clauses that constitute the sentences in those arguments – or, at least, constitute some plausible paraphrase or reformulation of those sentences.

We can see this if we return to this earlier argument:

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

We paraphrase the compound sentence 'Jenny is either happy or sad' into a compound of two canonical clauses, joined by the coordinator 'or':

› Jenny is happy or Jenny is sad.

And we paraphrase the noncanonical clause 'Jenny is not happy' as the negative clause

› It is not the case that: Jenny is happy.

Identifying the coordinator 'or' and the negative clause 'it is not the case that' as structural words, and replacing the canonical clause 'Jenny is happy' with the placeholder letter 'A', and the canonical clause 'Jenny is sad' with the placeholder letter 'B', we arrive again at the argument structure we previously identified:

A or B
not: A
So: B

In this and the other examples above, we removed all details of the arguments except for the special words 'and', 'or', 'not' and 'if ..., then ...', and replaced the other clauses in the argument (or its near paraphrase) by placeholder letters 'A', 'B', etc. (We were careful to replace the same clause by the same placeholder every time it appeared.)

Another example:

26. Butter isn't healthy, but it is delicious.

We begin by paraphrasing. We note that the pronoun 'it' is actually referring to the previously mentioned subject 'butter', and we move the negative 'isn't' into a position that reveals the canonical clause 'butter is healthy'. We also note that the effect of 'but' is roughly the same as our structural word 'and' (though it suggests a contrast that 'and' does not, 'but' expresses the roughly equivalent idea that each of the claims it connects are true) we obtain this more stilted paraphrase:

27. It is not the case that butter is healthy and butter is delicious.

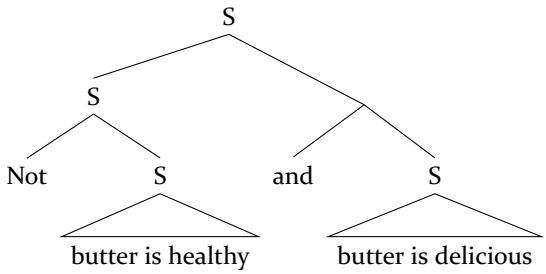


Figure 4.3: Paraphrase of Example 26 showing its subsentential structure, as in Example 27.

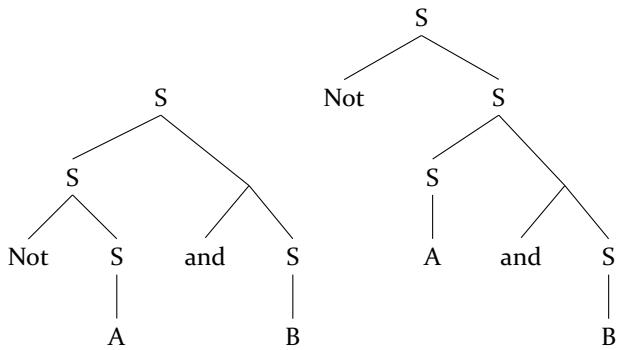


Figure 4.4: Different sentential structures for 'Not A and B' shown in schematic syntactic trees.

This paraphrase has the syntactic tree in Figure 4.3. Note that we have not further broken down the canonical clauses into subject-predicate form, because our structural words, the sentence connectives, do not occur within any canonical clause and hence are not 'visible' to our analysis at the level of sentences. Finally, we replace the canonical clauses with placeholder sentences, and we reach the structure 'Not A and B'.

The schematic paraphrase 'Not A and B' is potentially ambiguous, because it is not obvious whether the 'not' applies just to the A-clause, or to the whole 'A and B' clause. We could introduce parentheses to eliminate this ambiguity, distinguishing 'Not (A and B)' from 'Not A and B'. This is the approach we will take in our formal language Sentential(see page 41). Alternatively, we can use the hierarchical nature of syntactic trees to see see that there are two different structures possible for 'Not A and B', as depicted in the schematic syntactic trees in Figure 4.4.

Sharp-eyed readers will have noticed that our list of special words doesn't precisely line up with the canonical clauses we introduced in §4.2. Consider the noncanonical clause 'She said that Jenny is sad', in which the canonical clause 'Jenny is sad' is subordinate within an indirect speech report. Because 'She said that' is not on our special list of words, we cannot analyse this sentence as composed from a canonical clause and some

structural expression. So for the purposes of sentential logic, we will treat 'She said that Jenny is sad' *as if* it were canonical, even though it is not from the point of view of English grammar. From the point of view of our structural words, this sentence doesn't have any further structure that we can identify. Such QUASI-CANONICAL CLAUSES – clauses that do not feature any of our list of structural words – will also be replaced by placeholder letters in our analysis.

This raises another question. What makes the words on our list special? Logicians tend to take a pragmatic attitude to this question. They say: *nothing*! If you had chosen different words, you would have come up with a different structure. In terms of the syntactic trees we drew above, there is a hierarchy of levels when breaking down the top-level sentence into constituent phrases, and different choices of structural words correspond to different choices concerning the level at which to stop our analysis.

Linguists take a very expansive view of structural words, so that the class of canonical clauses is rather small, and there is a lot of structure to be identified in natural language. For example, the presence of modal auxiliaries (like 'will' or 'must', as in 'James must eat'), verb inflections other than the present tense (e.g., 'they took snuff'), and subordination (as in 'Etta knew that peaches were abundant'), in addition to items on our list of structural words, suffices to make a clause noncanonical. And there are logics which do take these to be structural words: modal logics, tense logics, and epistemic logics treat these as structural words and provide recipes for the structural analysis of arguments that abstract away features other than these words. But there is no fundamental principle that divides words once and for all into structural words and other words. So logicians are more interested in quasi-canonical clauses, given some fixed list of structural words, when using logic to model or represent natural languages.

We will start simply, and focus on the list of truth-functional sentence connectives, principally 'and', 'or', 'not', 'if ... then ...' and '... if and only if ...'. There are practical reasons why these words are useful ones to focus on initially, which I will now discuss.

4.4 Formal Languages

In logic, a FORMAL LANGUAGE is a language which is particularly suited to representing the structure or form of its sentences. Such languages have a precisely defined SYNTAX, that allows us to specify without difficulty the grammatical sentences of the language, and they also have a precise SEMANTICS which both tells us what meanings *are* in those languages, and assigns meanings to the sentences and their constituents.

Formal languages are not the same as natural languages, like English. The languages we will be looking at are much more limited in their expressive power than English, and not subject to ambiguity or imprecision in the way that English is. But a formal language can be very useful in *representing* or modelling features of natural language. In particular, they are very good at capturing the *structure* of natural language sentences and arguments, because we can design them specifically to represent a particular level of structural analysis. The semantics we give for such languages reflects that this is their primary use, as you will see. The languages assign fixed meanings only to those expressions which can be used to represent structure, but (unlike English) not every expression is treated as structural.

In this chapter we will begin developing a formal language which will allow us to represent many sentences of English, and arguments involving those sentences. The language will have a very small basic vocabulary, since we are designing it to represent the sentential structure of the examples with which we began. So it will have expressions corresponding to the English sentence connectives 'and', 'or', 'not' and 'if ..., then ...' and 'if and only if'. The language represents these English connectives by its own class of dedicated sentence connectives which allow simple sentences of the language to be combined into more complex sentences. These words are a good class to focus on in developing a formal language, because languages with expressions analogous to these feature a good balance between being useful and being well behaved and easy to study. The language we will develop is called Sentential, and the study of that language and its features is called *sentential logic*. (It also has other names: see Appendix A.)

Once we have our formal language Sentential, we will be able to go on (in chapter 3) to show that it has a very nice feature: it is able to represent the structure of a large class of valid natural language arguments. So while logic can't help us with every good argument, and it can't even help us with every conclusive argument, it can help us to understand arguments that are valid, or conclusive due to their structure, when that structure involves 'and', 'or', 'not', 'if' and various other expressions.

We will see later in this book that one could take additional expressions in natural language to be involved is determining the structure of a sentence or argument. The language we develop in chapter 4 is one that is suited to represent what we get when we take names like 'Juanita', predicates like 'is a vixen' or 'is a fox', and quantifier expressions like 'every' and 'some' to be structural – see also §16. And as we have mentioned, there are still other formal logical languages, unfortunately beyond the scope of this book, which take still other words or grammatical categories as structural: logics which take modal words like 'necessarily' or 'possibly' as structural, and logics which take temporal adverbs like 'always' and 'now' as structural.[3] What we notice here is that using logic to model natural language arguments inevitably involves a compromise. If we have lots of structural words, we can show many conclusive arguments to be valid, but our logic is complex and involves many different sentence connectives. If we take relatively few words as structural, we can represent fewer conclusive arguments as valid, but our formal language is a lot easier to work with.

We will start now by putting those tantalising observations about richer logics out of your mind! We'll focus to begin with on the language Sentential, and on how we can use it to model arguments in English featuring some particular sentence connectives as structural words, including those we've just highlighted above. This collection of structural words has struck many logicians over the years as providing a good balance between simplicity and strength, ideal for an introduction to logic.

[3] Some arguments will be valid in richer logical frameworks, but not according to the more austere framework for logical structure provided by Sentential. For example 'Sylvester is always active; so Sylvester is active now' is conclusive. From the point of view of temporal logic, the argument has this form 'Always A; so now A', which is valid in normal temporal logics. But it is not valid according to Sentential, because the premise 'Sylvester is always active' has no internal structural words that occur on the list of sentence connectives that Sentential represents.

4.5 Atomic Sentences

In §4.1, we started isolating the form of an argument by replacing quasi-canonical clauses (those including none of our list of structural words) within the argument with individual placeholder letters. Thus in the first example of this section, 'it is raining outside' is a quasi-canonical clause of 'If it is raining outside, then Jenny is miserable', and we replaced this clause with 'A'.

Our artificial language, Sentential, pursues this idea absolutely ruthlessly. We start with some ATOMIC SENTENCES, which are analogues of the canonical clauses of natural languages. These will be the basic building blocks out of which more complex sentences are built. Unlike natural languages, these simplest sentences of Sentential have no interesting internal structure at all. The atomic sentences of Sentential will consist of upper case italic letters, with or without numerical subscripts. There are only twenty-six letters of the alphabet, but there is no limit to the number of atomic sentences that we might want to consider, so we add numerical subscripts to each letter to extend our library of atomic sentences indefinitely. This procedure does give us atomic sentences with some syntax, but it is irrelevant to meaning: there is no connection between 'A_8' and 'A_{67}' just because they both involve 'A'. Likewise, there is no connection between 'D_{17}' and 'G_{17}' even though they share the same numerical subscript. Upper case italic letters, with or without subscripts, are all the atomic sentences there are. So, here are five different atomic sentences of Sentential:

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

Atomic sentences are the basic building blocks of Sentential. But what motivates their introduction is the fact that they can be used to represent, or *symbolise*, certain English sentences. To do this, we provide a SYMBOLISATION KEY, such as the following, which assigns a temporary linkage between some atomic sentences of Sentential, and some quasi-canonical sentences of the natural language we are representing using Sentential, in this case, English:

A: It is raining outside.
C: Jenny is miserable.

In doing this, we are not fixing this symbolisation *once and for all*. We are just saying that, for the time being, we shall use the atomic sentence of Sentential, 'A', to symbolise the English sentence 'It is raining outside', and the atomic sentence of Sentential, 'C', to symbolise the English sentence 'Jenny 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: Jenny is an anarcho-syndicalist.
C: Dipan is an avid reader of Tolstoy.

Given this flexibility, it isn't true that a sentence of Sentential **means the same thing** as any particular natural language sentence – see also §8.4. The question of what any

given atomic sentence of Sentential means is actually a bit hard to make sense of; we will return to it in §9.

But it is important to understand that whatever internal structure an English sentence might have is lost when it is symbolised by an atomic sentence of Sentential. From the point of view of Sentential, an atomic sentence is just a letter. It can be used to build more complex sentences, but it cannot be taken apart. So we cannot use Sentential to symbolise arguments whose validity depends on structure within sentences that are symbolised as atomic sentences of Sentential, such as arguments whose conclusiveness turns on different structural words than those Sentential seeks to capture, or arguments which depend on the internal subject-predicate structure of canonical clauses.

Key Ideas in §4

- Arguments are made of sentences, and to understand the structure of an argument we must understand the syntactic structure of those sentences, including analysing those sentences into their simplest constituents, roughly, how a compound sentence can be constructed by combining canonical clauses.
- The formal language Sentential is designed to model English arguments involving compound sentences structured by the sentence connectives 'and', 'or', 'not', and 'if', and related expressions.
- We symbolise these arguments this by abstracting away any other aspects of English sentences, using structureless atomic sentences to represent clauses that do not include these special expressions.
- Many English words and phrases could be treated as structural, and different formal languages can be motivated by other views about the structure of natural language sentences. But Sentential represents a particularly well behaved aspect of English sentence structure.

Practice exercises

A. True or false: if you are going to represent an English sentence by an atomic sentence in Sentential, the English sentence cannot have a sentence connective (like 'and') occurring within it.

B. Which one or more of the following are **not** atomic sentences of Sentential?

1. A';
2. W_0;
3. $Q_{5902222}$;
4. 77_P;
5. V^9.

C. This argument is invalid in English: 'There is water in the glass and it is cold; therefore it is cold'. Comment on why it is invalid, and what potential pitfalls arise when considering how to symbolise this argument into Sentential.

5

Connectives

In the previous section, we considered symbolising relatively simple English sentences with atomic sentences of Sentential, when they did not include any of our list of structural words, the English sentence connectives 'and', 'or', 'not', and so forth. This leaves us wanting to deal with sentences including these structural words. In Sentential, we shall make use of logical connectives to build complex sentences from atomic components. There are five logical connectives in Sentential, inspired by the English structural words we have encountered so far. This table summarises them, and they are explained throughout this section.

Symbol	What it is called	Rough English analogue
$\neg$	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 ...'

As the table suggests, we will introduce these connectives by the English connectives they parallel. It is important to bear in mind that they are perfectly legitimate standalone expressions of Sentential, with a meaning independent of the meaning of their English analogues. The nature of that meaning we will see in §9. Sentential is not a strange way of writing English, but a free-standing formal language, albeit one designed to represent some aspects of natural language.

5.1 Modelling and Paraphrase

We saw in §4.3 the grammatical heterogeneity of the sentence connectives. Even within the same class of connective we don't have grammatical uniformity. The different sentence connectives that express negation – e.g., 'not', '-n't', and 'it is not the case that' – can occur in quite different places in a grammatical sentence. Consider these examples:

1. Vassiliki doesn't like ballet;
2. Vassiliki dislikes ballet;
3. It is not the case that Vassiliki likes ballet.

To the grammarian, these differences are of great significance. To the logician, they are not. When the logician considers these examples, what matters is that these are all negated sentences, not the particular way that the idea of negation happens to be implemented syntactically in English. All these sentences seem to be expressing an idea that might roughly be expressed as 'It is not the case that: Vassiliki likes ballet'. These sentences are all acceptable PARAPHRASES of each other, because they all express more or less the same content. They need not be perfectly synonymous to be acceptable paraphrases, because sometimes the small divergences in meaning do not matter for our project.

Logic aims to represent relations within and between sentences that are significant for arguments. Since there are important grammatical differences that are argumentatively insignificant, logic will sometimes overlook linguistic accuracy in order to capture the 'spirit' of a sentence. That spirit is what is present in all acceptable paraphrases of that sentence. If some group of sentences can all play more or less the same role in an argument, the logician will aim to capture just what is essential to the argumentative role. In the following argument, it doesn't matter which of the previous examples we use for the second premise: the argument remains good whichever we include.

1. Vassiliki either likes ballet or soccer;
2. Vassiliki doesn't like ballet;

So: Vassiliki likes soccer.

Partly this tolerant attitude arises because logicians are interested principally in how to represent natural language arguments in a formal language. The formal language is already artificial and limited compared to the expressive power of natural language. Sentential only has five different sentence connectives, compared to the rich variety seen in English. From a logical point of view we are already sacrificing nuances of meaning when we represent an argument in a formal language. So it really doesn't matter which way of paraphrasing the original argument we choose, as long as it preserves the gist of the argument, because we'll already be modelling that argument in a way that cannot be faithful to its exact meaning. So when symbolising natural language arguments into English, it is generally appropriate to find a paraphrase that is good enough, but one that most explicitly displays the sentence connectives that you take to be involved in the logical structure of the argument. So while the sentence 'It is not the case that Vassiliki likes ballet' is much more stilted sounding than 'Vassiliki doesn't like ballet', it has the virtue of displaying the logical structure of the sentence more clearly. It is obvious, in the paraphrase, that we have a negation operating on the canonical clause 'Vassiliki likes ballet', and that makes symbolisation straightforward.

As will be evident throughout this section, symbolising an argument in Sentential is not like translating it into another natural language. Translation aims to preserve the meaning of your original argument in all its nuance. Symbolisation is more like *modelling*, where you choose to include certain important features and leave out other

features that are not important for you. A physicist, for example, might model a system of moving bodies by treating all of them as point particles. Of course the bodies are in fact extended in space, but for the purposes for which the model is designed it may be irrelevant to include that detail, if all that the physicist is trying to do is to predict the overall trajectories of those bodies. Likewise, in logic if our project is to analyse whether an argument is conclusive or not, we may not need to include every detail of meaning in order to complete that project. A good model needn't be perfectly accurate, and in fact, highly accurate models can be very poor because their additional complexity makes them too unwieldy to work with. We will happily settle for models which are good enough to represent the important details. In the present context, that means we want to paraphrase arguments in a way that makes their logical structure explicit, and then to symbolise them using the closest available Sentential connectives, even if they are not perfectly synonymous with the English sentence connectives they contain. We'll return to this issue further in §8.4.

Because a symbolisation isn't exactly alike in meaning to the original sentence, there is some room for the exercise of judgement. You will need sometimes to make choices between different paraphrases that seem to capture the gist of the sentence about equally well. You may have to make a *judgement* about what the sentence is 'really' trying to say. Such choices cannot be reduced to a mechanical algorithm. Though you can ask yourself some leading questions: 'which way was this ambiguous sentence intended?' or 'are these two options plausibly intended to be understood as mutually exclusive?'.

5.2 Negation

Consider how we might symbolise these sentences:

28. Mary is in Barcelona.
29. It is not the case that Mary is in Barcelona.
30. Mary is not in Barcelona.

In order to symbolise sentence 28, we will need an atomic sentence. We might offer this symbolisation key:

B: Mary is in Barcelona.

Since sentence 29 is obviously related to the sentence 28, we shall not want to symbolise it with a completely different sentence. Roughly, sentence 29 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 29 with '$\neg B$'.

Sentence 30 also contains the word 'not'. And it is obviously equivalent to sentence 29. As such, we can also symbolise it with '$\neg B$'.

It is much more common in English to see negation appear as it does in 30, with a 'not' found somewhere within the sentence, than the more formal form in 29. The form in 29 has the benefit that the sentence 'Mary is in Barcelona' itself appears as

a subsentence of 28 – so we can see how negation forms a new sentence by literally adding words to the old sentence. This is not so direct in 30. But since 29 and 30 are near enough synonymous in meaning, we can treat them both as negations of 28. (This issue is a bit tricky – see the discussion of examples 34 and 35.)

> If a sentence can be paraphrased by a sentence beginning 'It is not the case that ...', then it may be symbolised by $\neg\mathcal{A}$, where $\mathcal{A}$ is the Sentential sentence symbolising the sentence occurring at '...'.[1]

It will help to offer a few more examples:

31. The widget can be replaced.
32. The widget is irreplaceable.
33. The widget is not irreplaceable.

Let us use the following representation key:

R: The widget is replaceable.

Sentence 31 can now be symbolised by 'R'. Moving on to sentence 32: saying the widget is irreplaceable means that it is not the case that the widget is replaceable. So even though sentence 32 does not contain the word 'not', we shall symbolise it as follows: '$\neg R$'. This is close enough for our purposes.

Sentence 33 can be paraphrased as 'It is not the case that the widget is irreplaceable.' Which can again be paraphrased as 'It is not the case that it is not the case that the widget is replaceable'. So we might symbolise this English sentence with the Sentential sentence '$\neg\neg R$'. Any sentence of Sentential can be negated, not only atomic sentences, so '$\neg\neg R$' is perfectly acceptable as a sentence of Sentential. You might have the sense that these two negations should 'cancel out', and in Sentential that sense will turn out to be vindicated.

But some care is needed when handling negations. Consider:

34. Jane is happy.
35. Jane is unhappy.

If we let the Sentential-sentence 'H' symbolise 'Jane is happy', then we can symbolise sentence 34 as 'H'. However, it would be a mistake in general to symbolise sentence 35 with '$\neg H$'. If Jane is unhappy, then she is not happy; but sentence 35 does not mean the same thing as 'It is not the case that Jane is happy'. Jane might be neither happy nor unhappy; she might be in a state of blank indifference. In order to symbolise sentence 35, then, we will typically want to introduce a new atomic sentence of Sentential. Nevertheless, there may be limited circumstances where it doesn't matter which of these nonsynonymous ways to deny 'Jane is happy' I adopt.

Sometimes a sentence will include a negative for literary effect, even though what is expressed isn't negative. Consider

36. I don't like cricket; I love it!

The first clause seems negative, but the speaker confounds the hearer's expectation by going on to assert something even more positive than mere liking. Here the first clause really means something like 'I don't merely like cricket' – what is denied is that the speaker's affection for cricket is limited to mere liking.[2]

5.3 Conjunction

Consider these sentences:

37. Adam is athletic.
38. Barbara is athletic.
39. Adam is athletic, and Barbara is also athletic.

We will need separate atomic sentences of Sentential to symbolise sentences 37 and 38; perhaps

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

Sentence 37 can now be symbolised as 'A', and sentence 38 can be symbolised as 'B'. Sentence 39 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 39. 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 Sentential.

Some more examples will bring out this point:

40. Barbara is athletic and energetic.
41. Barbara and Adam are both athletic.
42. Although Barbara is energetic, she is not athletic.
43. Adam is athletic, but Barbara is more athletic than him.

Sentence 40 is obviously a conjunction. The sentence says two things (about Barbara). In English, it is permissible to refer to Barbara only once. It *might* be tempting to

[2] Technically, the negation here targets not the content of the clause, but the typical expectation that a cooperative speaker who says 'I like X' is communicating that liking is the highest point on the scale of affection their attitude to X reaches. These 'scalar implicatures' are discussed at length in Larry Horn (1989) *A Natural History of Negation*, University of Chicago Press, p. 382. We return to this 'metalinguistic' negation in §19.4 below.

think that we need to symbolise sentence 40 with something along the lines of 'B and energetic'. This would be a mistake. Once we symbolise part of a sentence as 'B', any further structure is lost. 'B' is an atomic sentence of Sentential. Conversely, 'energetic' is not an English sentence at all. What we are aiming for is something like 'B and Barbara is energetic'. So we need to add another sentence letter to the symbolisation key. Let 'E' symbolise 'Barbara is energetic'. Now the entire sentence can be symbolised as '$(B \wedge E)$'.

Sentence 41 says one thing about two different subjects. It says of both Barbara and Adam that they are athletic, and in English we use the word 'athletic' only once. The sentence can be paraphrased as 'Barbara is athletic, and Adam is athletic'. We can symbolise this in Sentential as '$(B \wedge A)$', using the same symbolisation key that we have been using.

Sentence 42 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 42 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 Sentential sentence '$(E \wedge \neg B)$'. Note that we have lost all sorts of nuance in this symbolisation. There is a distinct difference in tone between sentence 42 and 'Both Barbara is energetic and it is not the case that Barbara is athletic'. Sentential does not (and cannot) preserve these nuances.

Sentence 43 raises similar issues. There is a contrastive structure. The speaker who asserts it means something by that 'but' — something to the effect of there being a contrast between those two features. But their brief utterance doesn't tell exactly which contrast is intended. These contrasts are not something that Sentential is designed to deal with. So we can paraphrase the sentence as '*Both* Adam is athletic, *and* Barbara is more athletic than Adam'. (Notice that we once again replace the pronoun 'him' with 'Adam'.) How should we deal with the second conjunct? We already have the sentence letter 'A', which is being used to symbolise 'Adam 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 to symbolise the entire sentence, we need a new sentence letter. Let the Sentential sentence 'R' symbolise the English sentence 'Barbara is more athletic than Adam'. Now we can symbolise sentence 43 by '$(A \wedge R)$'.

> 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 I am putting parentheses around the conjunctions. The reason for this is to avoid potential ambiguity. This can be brought out by considering how negation might interact with conjunction. Consider:

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

Sentence 44 can be paraphrased as 'It is not the case that: both you will get soup and you will get salad'. Using this symbolisation key:

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

We would symbolise 'both you will get soup and you will get salad' as '$(S_1 \wedge S_2)$'. To symbolise sentence 44, then, we simply negate the whole sentence, thus: '$\neg(S_1 \wedge S_2)$'.

Sentence 45 is a conjunction: you *will not* get soup, and you *will* get salad. 'You will not get soup' is symbolised by '$\neg S_1$'. So to symbolise sentence 45 itself, we offer '$(\neg S_1 \wedge S_2)$'.

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. Parentheses help us to avoid ambiguity by clear distinguishing these two cases.

Once again, however, English does feature this sort of ambiguity. Suppose instead of 44, we'd just said

46. You won't get soup with salad.

The sentence 46 is arguably ambiguous; one of its readings says the same thing as 44, the other says the same thing as 45. Parentheses enable Sentential to avoid precisely this ambiguity.

The introduction of parentheses prompts us to define some other useful concepts. If a sentence of Sentential has the overall form $(\mathcal{A} \wedge \mathcal{B})$, we say that its *main connective* is conjunction – even if other connectives occur within $\mathcal{A}$ or $\mathcal{B}$. Likewise, if the sentence has the form $\neg\mathcal{A}$, its main connective is negation. We define the *scope* of an occurrence of a connective in a sentence as the subsentence which has that connective as its main connective. So the scope of '$\neg$' in '$\neg(S_1 \wedge S_2)$' is the whole sentence (because negation is the main connective), while the scope of '$\neg$' in '$(\neg S_1 \wedge S_2)$' is just the subsentence '$\neg S_1$' – the main connective of the whole sentence is conjunction. Parentheses help us keep track of things the scope of our connectives, and which connective in a sentence is the main connective. I say more about the notions of a main connective and the scope of a connective in §6.3.

5.4 Disjunction

Consider these sentences:

47. Either Denison will play golf with me, or he will watch movies.
48. 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 shall again need to introduce a new symbol. Sentence 47 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 48 is only slightly more complicated. There are two subjects, but the English sentence only gives the verb once. However, we can paraphrase sentence 48 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 $(\mathcal{A} \vee \mathcal{B})$ 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, 'Entrees come with either soup or salad': you may have soup; you may have salad; but, if you want *both* soup *and* salad, then you have to pay extra.

At other times, the word 'or' allows for the possibility that both disjuncts might be true. This is probably the case with sentence 48, above. I might play golf with Denison, with Ellery, or with both Denison and Ellery. Sentence 48 merely says that I will play with *at least* one of them. This is called an INCLUSIVE OR. The Sentential symbol '$\vee$' always symbolises an inclusive 'or'.

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

49. Either you will not have soup, or you will not have salad.
50. You will have neither soup nor salad.
51. You get either soup or salad, but not both.

Using the same symbolisation key as before, sentence 49 can be paraphrased in this way: 'Either *it is not the case that* you get soup, or *it is not the case that* you get salad'. To symbolise this in Sentential, we need both disjunction and negation. 'It is not the case that you get soup' is symbolised by '$\neg S_1$'. 'It is not the case that you get salad' is symbolised by '$\neg S_2$'. So sentence 49 itself is symbolised by '$(\neg S_1 \vee \neg S_2)$'.

Sentence 50 also requires negation. It can be paraphrased as, '*It is not the case that* either you get soup or you get salad'. Since this negates the entire disjunction, we symbolise sentence 50 with '$\neg(S_1 \vee S_2)$'.

Sentence 51 is an exclusive 'or'. We can break the sentence into two parts. The first part says that you get one or the other. We symbolise this as '$(S_1 \vee S_2)$'. The second part says that you do not get both. We can paraphrase this as: 'It is not the case both that you get soup and that you get salad'. Using both negation and conjunction, we symbolise this with '$\neg(S_1 \wedge S_2)$'. Now we just need to put the two parts together. As we saw above, 'but' can usually be symbolised with '$\wedge$'. Sentence 51 can thus be symbolised as '$((S_1 \vee S_2) \wedge \neg(S_1 \wedge S_2))$'. This last example shows something important. Although the Sentential symbol '$\vee$' always symbolises inclusive 'or', we can symbolise an exclusive 'or' in Sentential. We just have to use a few of our other symbols as well.

5.5 Conditional

Consider these sentences:

52. If Jean is in Paris, then Jean is in France.
53. Jean is in France only if Jean is in Paris.

Let's use the following symbolisation key:

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

Sentence 52 is roughly of this form: 'if P, then F'. We will use the symbol '→' to symbolise this 'if ..., then ...' structure. So we symbolise sentence 52 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 53 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 52. That would be a mistake. My knowledge of geography tells me that sentence 52 is unproblematically true: there is no way for Jean to be in Paris that doesn't involve Jean being in France. But sentence 53 is not so straightforward: were Jean in Dijon, Marseilles, or Toulouse, Jean would be in France without being in Paris, thereby rendering sentence 53 false. Since geography alone dictates the truth of sentence 52, whereas travel plans (say) are needed to know the truth of sentence 53, they must mean different things.

In fact, sentence 53 can be paraphrased as 'If Jean is in France, then Jean 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 'A only if B' or 'B if A'.

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

54. For Jean to be in Paris, it is necessary that Jean be in France.
55. It is a necessary condition on Jean's being in Paris that she be in France.
56. For Jean to be in France, it is sufficient that Jean be in Paris.
57. It is a sufficient condition on Jean's being in France that she be in Paris.

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

It is important to bear in mind that the connective '→' 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 seem to lose a huge amount when we use '→' to symbolise English conditionals. We shall return to this in §§8.6 and 11.5.

5.6 Biconditional

Consider these sentences:

58. Shergar is a horse only if it he is a mammal.
59. Shergar is a horse if he is a mammal.
60. 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.

Sentence 58, for reasons discussed above, can be symbolised by '$H \to M$'.

Sentence 59 is importantly different. It can be paraphrased as, 'If Shergar is a mammal then Shergar is a horse'. So it can be symbolised by '$M \to H$'.

Sentence 60 says something stronger than either 58 or 59. 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 58 and 59. So we can symbolise it as '$(H \to M) \wedge (M \to H)$'. 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 Sentential symbol to deal with exclusive 'or', we do not really need a new Sentential symbol to deal with biconditionals. However, we will use '$\leftrightarrow$' to symbolise the biconditional. So we can symbolise sentence 60 with the Sentential sentence '$H \leftrightarrow M$'.

The expression 'if and only if' occurs a lot in philosophy and logic. For brevity, we can abbreviate it with the snappier word 'IFF'. I shall 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 as 'A iff B'; that is, as 'A if and only if B'.

Other expressions in English which can be used to mean 'iff' include 'exactly if' and 'exactly when', or even 'just in case'. So if we say 'You run out of time exactly when the buzzer sounds', we mean: 'if the buzzer sounds, then you are out of time; and also if you are out of time, then the buzzer sounds'.

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 vegetables, you won't get any dessert'. Suppose you ate your vegetables, but that your parents refused to give you any dessert, on the grounds that they were only committed to the *conditional* (roughly 'if you get dessert, then you will have eaten your vegetables'), rather than the biconditional (roughly, 'you get dessert iff you eat your vegetables'). 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.

5.7 Unless

We have now introduced all of the connectives of Sentential. We can use them together to symbolise many kinds of sentences, but not every kind. It is a matter of judgement whether a given English connective can be symbolised in Sentential. One rather tricky case is the English-language connective 'unless':

61. Unless you wear a jacket, you will catch cold.
62. 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.

How should we try to symbolise these in Sentential? Note that 62 seems to say: 'either you will catch cold, or if you don't catch cold, it will be because you wear a jacket'. That would have this symbolisation in Sentential: '$(D \lor (\neg D \to J))$'. This turns out to be just a long-winded way of saying '$(\neg D \to J)$', i.e., if you don't catch cold, then you will have worn a jacket.

Equally, however, 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 '$\neg J \to D$'.

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 '$J \lor D$'.

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

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

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.

We should not take 'unless' to always have the stronger biconditional form. Consider this example:

63. We'll capture the castle, unless the Duke tries to stop us.

This certainly says that if we fail to capture the castle, it will have been because of that pesky Duke. But what if the Duke tries *but fails* to stop us? We might in that case capture the castle even though he tried to stop us. While 63 is still true, it is not the case that 'if the Duke tries to stop us, we won't capture the castle' is true.

Key Ideas in §5

- Sentential features five connectives: '∧' ('and'), '∨' ('or'), '¬' ('not'), '→' ('if ..., then ...') and '↔' ('if and only if' or 'iff').
- These connectives, alone and in combination, can be used to symbolise many English constructions, even some which do not feature the English counterparts of the connectives – as when we approached the symbolisation of sentences involving 'unless'.
- Figuring out how to symbolise a given natural language sentence might not be straightforward, however, as in the cases of 'A if B' and 'A only if B', which have quite different symbolisations into Sentential. 'A unless B' is perhaps even trickier, being sometimes used ambiguously by English speakers and being able to be symbolised in many good ways.
- What matters in symbolisation is that the 'spirit' of the argument is preserved and modelled appropriately, not that every nuance of meaning is preserved.

Practice exercises

A. Using the symbolisation key given, symbolise each English sentence in Sentential.

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 Sentential.

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 Sentential.

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, 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 Sentential.

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 Sentential.

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 Sentential.

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 '∨', '∧', and '¬'. How could you symbolise an exclusive 'or' using only two connectives? Is there any way to symbolise an exclusive 'or' using only one connective?

6

Sentences of Sentential

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

6.1 Expressions

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

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	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
Parentheses	(,)

We define an EXPRESSION of Sentential as any finite nonempty string of symbols of Sentential. Take any of the symbols of Sentential and write them down, in any order, and you have an expression of Sentential. Expressions are sometimes called 'strings', because they are just a string of symbols from the approved list above. No restriction is placed on expressions apart from having to contain at least one character, and not going on infinitely long.

6.2 Sentences

We want to know when an expression of Sentential amounts to a *sentence*. Many expressions of Sentential will be uninterpretable nonsense. '$)A_{17}JQF\neg K))\wedge)()$' is a perfectly

good expression, but doesn't look likely to end up a correctly formed sentence of our language. Accordingly, we need to clarify the grammatical rules of Sentential. We've already seen some of those rules when we introduced the atomic sentences and connectives. But I will now make them all explicit.

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 Sentential, $\neg\mathcal{A}$ is a sentence of Sentential. (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 Sentential, then $(\mathcal{A} \wedge \mathcal{B})$ is a sentence of Sentential. Providing clauses like this for all of the connectives, we arrive at the following formal definition for a sentence of Sentential

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 Sentential. Just as the recursive definition allows complex sentences to be built up from simpler parts, the definition allows us to decompose sentences into their simpler parts. And if we get down to atomic sentences, then we are ok.

Let's consider some examples.

1. Suppose we want to know whether or not '$\neg\neg\neg D$' is a sentence of Sentential. Looking at the second clause of the definition, we know that '$\neg\neg\neg D$' is a sentence *if* '$\neg\neg D$' is a sentence. So now we need to ask whether or not '$\neg\neg D$' is a sentence. Again looking at the second clause of the definition, '$\neg\neg D$' is a sentence *if* '$\neg D$' is. Again, '$\neg D$' is a sentence *if* 'D' is a sentence. Now 'D' is an atomic sentence of Sentential, so we know that 'D' is a sentence by the first clause of the definition. So for a compound sentence like '$\neg\neg\neg D$', we must apply the definition repeatedly. Eventually we arrive at the atomic sentences from which the sentence is built up.

2. Next, consider the expression '$\neg(P \wedge \neg(\neg Q \vee R))$'. Looking at the second clause of the definition, this is a sentence if '$(P \wedge \neg(\neg Q \vee R))$' is. And this is a sentence if *both* 'P' *and* '$\neg(\neg Q \vee R)$' are sentences. The former is an atomic sentence, and the latter is a sentence if '$(\neg Q \vee R)$' is a sentence. It is. Looking at the fourth clause of the definition, this is a sentence if both '$\neg Q$' and 'R' are sentences. And both are!

3. Suppose we want to know whether '$(A\neg \vee B^1)$' is a sentence. By the third clause of the definition, this is a sentence iff its two constituents are sentences. The second is not: it consists of an upper case letter and a numerical *superscript*, which is not in conformity with the definition of an atomic sentence. The first isn't either: while it is constructed from an upper case letter and the negation symbol, they are in the wrong order to be a sentence by the first clause of the definition. So the original expression is not a sentence.

4. A final example. Consider the expression '$(P \rightarrow \neg(Q \rightarrow P)$'. If this is a sentence, then it's main connective is '$\rightarrow$', and it was formed from the sentences 'P' and '$\neg(Q \rightarrow P$'. 'P' is a sentence because it is a sentence letter. Is '$\neg(Q \rightarrow P$' a sentence? Only if '$(Q \rightarrow P$' is a sentence. But this isn't a sentence; it lacks a closing parenthesis which would need to be there if this was correctly formed using the clause in the definition covering conditionals. It follows that the expression with which we started isn't a sentence either.

6.3 Main Connectives and Scope

Ultimately, every sentence of Sentential is constructed in a predictable way 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 CONNECTIVE of the sentence. In the case of '$\neg\neg\neg D$', the main connective is the very first '$\neg$' sign. In the case of '$(P \wedge \neg(\neg Q \vee R))$', the main connective is '$\wedge$'. In the case of '$((\neg E \vee F) \rightarrow \neg\neg G)$', the main connective is '$\rightarrow$'.

The recursive structure of sentences in Sentential 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 Sentential also allows us to give a formal definition of the *scope* of a negation (mentioned in §5.3). The scope of '$\neg$' is the subsentence for which '$\neg$' is the main connective. So a sentence like

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

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 connective. So the scope of the negation is just '$\neg(R \wedge B)$'. More generally:

> The SCOPE of an instance of a connective (in a sentence) is the subsentence which has that instance of the connective as its main connective.

I talk of 'instances' of a connective because, in an example like '$\neg\neg A$', there are two occurrences of the negation connective, with different scopes – one is the main connective of the whole sentence, the other has just '$\neg A$' as its scope.

The recursive definition of a sentence of Sentential can also be depicted using a FORMATION TREE, similar to the syntactic trees for English we saw in §4, but much simpler. At each leaf node of the tree is either an atomic sentence or a sentence connective. Each nonleaf node contains a Sentential sentence, and branching from it are (i) its main connective, and (ii) the immediate subsentences in the scope of the main connective. Sentential sentences are just those expressions of the language that have a formation tree respecting these rules. Consider again the sentence '$\neg(P \wedge \neg(\neg Q \vee R))$'. This has the formation tree depicted in Figure 6.1.

At the beginning of this chapter, we introduced the sentences of Sentential in a way that was parasitic upon identifying the structure of certain English sentences. Now we can see that Sentential has its own syntax, which can be understood and used independently of English. Our understanding of the meaning of the sentence connectives is still tied to their English counterparts, but in §8.3 we will see that we can also understand their meaning independently from the natural language we used to motivate them. However, it remains true that what makes Sentential *useful* is that its syntax and connectives are designed to capture, more or less, elements of the structure of natural language sentences.

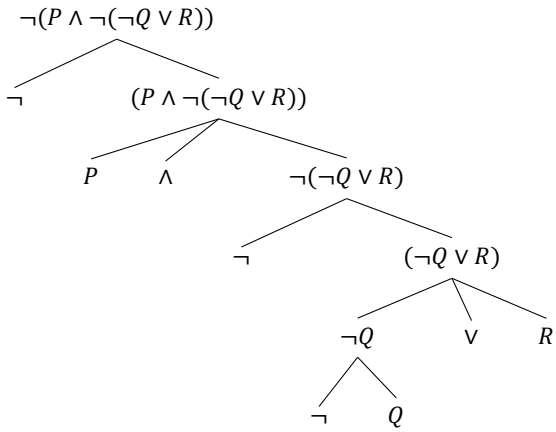


Figure 6.1: Formation tree for '$\neg(P \wedge \neg(\neg Q \vee R))$'.

6.4 Structure and Ambiguity

In later sections, we will see the utility of symbolising arguments to evaluate their validity. But even at this early stage, there is value in symbolising sentences, because doing so can clear up *ambiguity*.

Consider the following imagined regulation: 'Small children must be quiet and seated or carried at all times'. This sentence has an ambiguous structure, by which I mean: there are two different grammatical structures that it might be thought to have. We can draw up a syntactic tree to show this. But equally, symbolising can bring this out. Let us use the following key:

Q: Small children must be quiet.
S: Small children must be seated.
C: Small children must be carried.

Leaving the temporal adverbial phrase 'at all times' aside, here are two symbolisations of our target sentence:

Q and either S or C – in Sentential, $(Q \wedge (S \vee C))$;
either Q and S, or C – in Sentential, $((Q \wedge S) \vee C)$.

This ambiguity matters. If a child is carried, but is not quiet, the parent is violating the regulation with the first structure, but in compliance with the regulation with the second structure. As we will soon see, Sentential has resources to ensure that no ambiguity is present in any symbolisation of a given target sentence, which helps us make clear what we really might have meant by that sentence. And as this case makes clear, that might be important in the framing of legal statutes or contracts, in the formulation of government policy, and in other written documents where clarity of meaning is crucial.

6.5 Parenthetical Conventions

Strictly speaking, the parentheses in '$(Q \wedge R)$' are an indispensable part of the sentence. Part of this is because we might use '$(Q \wedge R)$' as a subsentence in a more complicated sentence. For example, we might want to negate '$(Q \wedge R)$', obtaining '$\neg(Q \wedge R)$'. If we just had '$Q \wedge R$' without the parentheses and put a negation in front of it, we would have '$\neg Q \wedge R$'. It is most natural to read this as meaning the same thing as '$(\neg Q \wedge R)$'. But as we saw in §5.3, 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 Sentential, however, it will make our lives easier if we are sometimes a little less than strict. So, here are some convenient conventions.

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

Second, it can be a bit painful to stare at long sentences with many nested pairs of parentheses. To make things a bit easier on the eyes, we shall allow ourselves to use parentheses in varied sizes, which sometimes helps in seeing which parentheses pair up with each other.

Combining these two conventions, we can rewrite the unwieldy sentence

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

rather more readably as follows:

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

The scope of each connective is now much clearer.

There are systems of logic which omit parentheses, using so-called 'Polish notation'. I discuss this notation briefly in Appendix A, p. 344.

Key Ideas in §6

- The class of sentences of Sentential has a perfectly precise recursive definition that allows us to determine in a step-by-step fashion, for any expression, whether it is a sentence or not.
- The main connective of a sentence is the final rule applied in the construction of a sentence; the scope of a connective is that subsentence in the construction of which it is the main connective.
- Each Sentential sentence, unlike English sentences, has an unambiguous structure.
- We can sometimes permit ourselves some liberality in the use of parentheses in Sentential, when we can be sure it gives rise to no difficulty.

Practice exercises

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

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

B. Construct a formation tree in the style of Figure 6.1 for the following sentences:

1. $(((A \to B) \land (B \to A)) \to (B \leftrightarrow A))$
2. $(((P \to Q) \land (\neg R \to Q)) \lor \neg R)$.

C. Are there any sentences of Sentential that contain no atomic sentences? Explain your answer.

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

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

7

Use and Mention

In this chapter, I have talked a lot *about* sentences. So I need to pause to explain an important, and very general, point.

7.1 Quotation Conventions

Consider these two sentences:

- Malcolm Turnbull is the Prime Minister.
- The expression 'Malcolm Turnbull' is composed of two upper case letters and thirteen lower case letters.

When we want to talk about this ex-Prime Minister, we USE his name. When we want to talk about his name, we MENTION that name. And in English, we normally 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 typically have to *mention* those words.[1] We need to indicate that we are mentioning them, rather than using them. To do this, some convention is needed. We can surround the expression in matched left and right quotation marks, or display them centrally in the page (say). So this sentence:

- 'Malcolm Turnbull' is the Prime Minister.

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

[1] More generally, when we want to talk about something we use its name. So when we want to talk about an expression, we use the name of the expression – which is just the expression enclosed in quotation marks. Mentioning an expression is using a name of that expression.

› Malcolm Turnbull is composed of two upper case letters and thirteen lower case letters.

also says something false: Malcolm Turnbull is a man, made of meat rather than letters. One final example:

› ‘‘Malcolm Turnbull’’ is the name of ‘Malcolm Turnbull’.

On the left-hand-side, here, we have the name of a name (it consists of an expression in quotation marks, and that embedded expression itself contains quotation marks). On the right hand side, we have a name (of an expression). 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, Sentential, and so we are often *mentioning* expressions from Sentential.

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: Sentential. The metalanguage is English. Not conversational English exactly, but English supplemented with some additional vocabulary which helps us to get along.

Now, I have used italic upper case letters for atomic sentences of Sentential:

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

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

› D is an atomic sentence of Sentential.

Obviously, I am trying to come out with an English sentence that says something about the object language (Sentential). But ‘D’ is a sentence of Sentential, 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 Sentential.

The general point is that, whenever we want to talk in English about some specific expression of Sentential, 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.

English is, generally, its own metalanguage. An expression of English enclosed in matching quotation marks is another expression of English, as the quotation marks are parts of English too. This causes a potential problem of ambiguity if the expression quoted itself contains quotation marks. English allows us to talk about operations on English expressions, as in this example

64. An English word results from adding ‘ing’ or ‘ion’ to the expression ‘confus’.

But this example is ambiguous. On one reading, it is discussing the expressions ‘confusing’ and ‘confusion’, and saying truly that they are both English words. But on another reading, the matching quotation marks are the one before ‘ing’ and the one after ‘ion’, and Example 64 is stating falsely that this unusual string of English letters and punctuation can be added to ‘confus’ to form an English word:

ing’ or ‘ion

To avoid this, we might introduce some mechanism for indicating which quotation marks are matched with each other.

7.3 Script Fonts, and Recursive Definitions Revisited

However, we do not just want to talk about *specific* expressions of Sentential. We also want to be able to talk about *any arbitrary* sentence of Sentential. Indeed, I had to do this in §6, when I presented the recursive definition of a sentence of Sentential. I used upper case script font letters to do this, namely:

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

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

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

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

To come at this distinction a slightly different way, while '$\mathcal{A}$' designates a sentence of Sentential, it can designate a different sentence on different occasions. It behaves a bit a like a pronoun. The pronoun 'it' always designates some object, but a different one in different circumstances of use. Likewise '$\mathcal{A}$' can stand for different sentences of Sentential. By contrast, 'A' always names just one atomic sentence of Sentential, the first letter of the English alphabet.

This last example raises a further complication for our quotation conventions. I have not included any quotation marks in the clauses of our recursive definition of a sentence of Sentential in §6.2. Should I have done so?

The problem is that the expression on the right-hand-side of most of our recursive clauses are not sentences of English, since they contain Sentential connectives, like '$\neg$'. Consider clause 2. We might try to write:

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

But this is no good: '$\neg\mathcal{A}$' is not a Sentential sentence, since '$\mathcal{A}$' is a symbol of (augmented) English rather than a symbol of Sentential. What we really want to say is something like this:

2″. If $\mathcal{A}$ is any Sentential sentence, then the expression that consists of the symbol '$\neg$', followed immediately by the sentence $\mathcal{A}$, is also a sentence.

This is impeccable, but rather long-winded. But we can avoid long-windedness by creating our own conventions. We can perfectly well stipulate that an expression like '$\neg\mathcal{A}$' should simply be read as abbreviating the long-winded account. So, *officially*, the metalanguage expression '$\neg\mathcal{A}$' simply abbreviates:

the expression that consists of the symbol '$\neg$' followed by the sentence $\mathcal{A}$

and similarly, for expressions like '$(\mathcal{A} \wedge \mathcal{B})$', '$(\mathcal{A} \vee \mathcal{B})$', etc. The latter is the expression which consists of an opening parenthesis, followed by the sentence $\mathcal{A}$, followed by the symbol '$\vee$', followed by the sentence $\mathcal{B}$, followed by a closing parenthesis.

If you like, you can think of our recursive definition of a sentence as a schema standing for infinitely many instances of each clause, one for each Sentential sentence. In the schematic clause for negation ('If $\mathcal{A}$ is a sentence, $\neg\mathcal{A}$ is also a sentence'), we can

consider each instance involving '$\mathcal{A}$' being replaced by some Sentential sentence surrounded by quotation marks in accordance with our conventions. So '$\neg\mathcal{A}$' is to be understood as abbreviating the expression consisting of a left quotation mark, a negation sign, the same Sentential sentence as $\mathcal{A}$, and a right quotation mark. Hence if if $\mathcal{A}$ is 'P', $\neg\mathcal{A}$ just is '$\neg P$'.

7.4 Quotation Conventions for Arguments

One of our main purposes for using Sentential is to study arguments, and that will be our concern in chapter 3. In English, the premises of an argument are often expressed by individual sentences, and the conclusion by a further sentence. Since we can symbolise English sentences, we can symbolise English arguments using Sentential. Thus we might ask whether the argument whose premises are the Sentential sentences 'A' and '$A \to C$', and whose conclusion is the Sentential sentence 'C', is valid. However, it is quite a mouthful to write that every time. So instead I shall introduce another bit of abbreviation. This:

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

abbreviates:

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

To avoid unnecessary clutter, we shall not regard this as requiring quotation marks around it. This is a name of an argument, not an argument itself. (Note, then, that '$\therefore$' is a symbol of our augmented *metalanguage*, and not a new symbol of Sentential.)

7.5 Pedantry in Practice

Having been precise about use and mention, you can now relax! If you've understood this section, you know how to do things properly. In exercises and practice problems, unless explicit instructions otherwise are given, you will be expected to do things properly and respect the distinction between use and mention. But your understanding of the topic means that you can probably do things a bit more sloppily elsewhere – safe in the knowledge you can fix them up if you need to. As the great twentieth century philosopher David Lewis said of the way he presented his account of the word 'knows' in his paper 'Elusive Knowledge',[2]

> I could have said my say fair and square, bending no rules. It would have been tiresome, but it could have been done.... I could have taken great care to distinguish between (1) the language I use when I talk about knowledge, or whatever, and (2) the second language that I use to talk about the semantic and pragmatic workings of the first language. If you want to hear my story told that way, you probably know enough to do the job for yourself.

[2] David Lewis (1996) 'Elusive Knowledge', *Australasian Journal of Philosophy* **74**, pp. 549–67, at pp. 566–7.

Wise words. In the end, the distinction between use and mention is intended to remove potential confusion. But sometimes over-eager application of it can prove just as big an obstacle to communication.

Key Ideas in §7

- It is crucial to distinguish between use and mention – between talking about the world, and talking about expressions.
- We use Sentential to represent sentences and arguments. But we use English – augmented with some additional vocabulary – to talk about Sentential.
- We introduced a slightly unusual convention for understanding quoted expressions involving script font letters: '$(\mathcal{A} \rightarrow \mathcal{B})$', to take a representative example, is to be interpreted as the Sentential expression consisting of a left parenthesis, followed by whatever Sentential expression $\mathcal{A}$ represents, followed by '$\rightarrow$', followed by whatever Sentential expression $\mathcal{B}$ represents, followed by a closing right parenthesis.

Practice exercises

A. For each of the following: Are the quotation marks correctly used, strictly speaking? If not, propose a corrected version.

1. Snow is not a sentence of English.
2. '$\mathcal{A} \rightarrow \mathcal{C}$' is a sentence of Sentential.
3. '$\neg\mathcal{A}$' is the expression consisting of the symbol '$\neg$' followed by the upper case script letter 'A'.
4. If '$\mathcal{A}$' is a sentence, so is '$(\mathcal{A} \vee \mathcal{A})$'.
5. '$\mathcal{A}$' has the same number of characters as 'A'.

B. Example 64 was ambiguous because it was unclear which pairs of quotation marks were matched with each other. Can you come up with a proposal for how we might indicate matching quotation marks to avoid this potential ambiguity?

Chapter 3

Truth Tables

8

Truth-Functional Connectives

8.1 Functions

So much for the grammar or syntax of Sentential. We turn now to the meaning of Sentential sentences. For technical reasons, it is best to start with the intended interpretation of the connectives.

As a preliminary, we need to have the concept of a (mathematical) *function*. Frequently, we refer to things not by name, but by the relations they have to other things. You can refer to Barack Obama by name, but one could equally refer to him in relation to his role, as 'the 44th President of the United States', or in relation to his family, as 'the husband of Michelle and father of Malia and Sasha'. These kinds of referring expressions are known as *descriptions*, and we will look at them in more detail in §19. Our interest in them now is in the relations they involve. Barack Obama is denoted by 'the biological father of Malia Obama' or 'the biological father of Sasha Obama', in relation to his children. But we can consider 'the biological father of ...' in relation to other individuals: so 'the biological father of Ivanka Trump' denotes Donald, 'the biological father of Daisy Turnbull' denotes Malcolm, and so on. We can summarise this information in a table like this:

Input	'the biological father of ...'
Malia Obama	Barack
Sasha Obama	Barack
Ivanka Trump	Donald
Daisy Turnbull	Malcolm
...	...

In this table we have an *input*, on the left, which is related to the *output* on the right. The relation which maps the things in the left column to their corresponding outputs on the right is known as a function – in this case, the 'biological father of' function. More precisely, a FUNCTION is a relation between the members of some collection A and

some collection B (which may be the same as A), such that each input to the function is a member of A, each output of the function is a member of B, and, crucially, each member of A is associated with at most one member of B. So 'the biological father of ...' is a function from the set of people (living or dead) to itself, and associates each person with their biological father. We implicitly that 'biological father' permits each person to be associated with a unique father. If we consider other notions of fatherhood, such as *paternal figure*, those would not yield a function, because many people have two or more paternal figures in their lives. Note that while everyone is associated with a unique biological father by this function (no input is associated with more than one output), the converse does not hold: some outputs are linked to more than one input, in fact, every father with more than one child.

Common examples of functions occur in mathematics: we can consider the function 'the sum of x and y', which takes two numbers as input, and spits out their unique sum, $x + y$. This is again a function from a set to itself, this time the set of integers. There are functions which are from one set to another: consider, 'the number of ...'s children', which is a function from people to numbers, mapping each person to the number of children they have. (Even if they have more than one child, there is still a unique number characterising how many they have, and that is what this function spits out.) There are functions which do not associate an output with every input: consider 'the eldest child of', which associates each parent with their first-born child, but doesn't associate nonparents with anything. There are many relations which are not functions. While Barack Obama can be characterised as 'the father of Malia', Malia Obama cannot be characterised as 'the child of Barack', since that attempted description would apply equally to her sister.

8.2 The Idea of Truth-Functionality

The relevance of the notion of a function is that we are going to identify the meanings of the Sentential sentence connectives with a certain class of functions.

A valid argument is one with a structure that guarantees the truth of the conclusion, given the truth of the premises (§2.5). Our interest in valid arguments leads us to be interested in the truth or falsity of sentences. Sentential gives rules governing the construction of complex sentences from smaller constituents for each sentence connective. The meanings of the sentence connectives in Sentential are likewise going to allow us to 'construct', or determine, the truth-value of a complex sentence as the result of the truth values of its constituent sentences and the main connective of that sentence.

This is an important idea about Sentential sentence connectives. They are each associated with a rule that fixes the truth value of a complex sentence of which they are the main connective, given the truth values of the constituent sentences. But such a rule is just a function: a function that takes one or two truth values as input, and yields a truth value as output. Such a function is called a TRUTH-FUNCTION. We can now summarise our important insight about Sentential:

A connective is TRUTH-FUNCTIONAL iff the truth value of a sentence with that connective as its main connective is uniquely determined by the truth value(s) of the constituent sentence(s), i.e., its meaning is a truth-function.
Every connective in Sentential is truth-functional.

It turns out that we don't need to know anything more about the atomic sentences of Sentential than their truth values to assign a truth value to those nonatomic, or COMPOUND, sentences. More generally, the truth value of any compound sentence depends only on the truth value of the subsentences that comprise it. In order to know the truth value of '$(D \wedge E)$', for instance, you only need to know the truth value of 'D' and the truth value of 'E'. In order to know the truth value of '$\big((D \wedge E) \vee F\big)$', you need only know the truth value of '$(D \wedge E)$' and 'F'. And so on. This is fact a good part of the reason why we chose these connectives, and chose their English near-equivalents as our structural words. To determine the truth value of some Sentential sentence, we only need to know the truth value of its components. This is why the study of Sentential is termed *truth-functional logic*.

8.3 Schematic Truth Tables

We introduced five connectives in chapter 2. To give substance to the claim that they are truth-functional, we simply need to explain how each connective yields a truth value for a compound sentence with that connective as its main connective, when supplied with the truth values of its IMMEDIATE SUBSENTENCES. In any sentence of the form $\mathcal{A} \rightarrow \mathcal{B}$, $\mathcal{A}$ and $\mathcal{B}$ are the immediate subsentences, since those combine with the main connective to form the sentence.

Not only does the truth value of a compound sentence of Sentential depend only on the truth values assigned to its subsentences, it does so *uniquely*, because it is a function. We may happily refer to *the* truth value of $\mathcal{A}$ as determined by its constituents. This means we may represent the pattern of dependence of compound sentence truth values on immediate subsentence truth values in a simple table, like those we saw in §8.1. These tables show how to relate the input truth values to output truth values for any sentence connective. For convenience, we shall represent the truth value True by 'T' and the value False by 'F'. (Just to be clear, the two truth values are True and False; the truth values are not *letters*!)

These truth tables completely characterise how the connectives of Sentential behave. Accordingly, we can take these schematic truth tables as giving the *meanings* of the connectives. The truth function associated with a connective captures their entire contribution to the sentences in which they appear.

A language, natural or artificial, is COMPOSITIONAL iff the meaning of a complex sentence is the product of the syntactic structure of the sentence and the meanings of its constituents. In Sentential, the only dimension of meaning is truth value. So in Sentential, we have a special case of compositionality: the truth value of a complex sentence depends on the syntax and the truth value of the constituent atomic sentences.

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 dependence in the SCHEMATIC TRUTH TABLE for negation, which shows how any sentence with negation as its main connective has a truth value depending on the truth value of its immediate subsentence:

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

This is a schematic table, because the input truth values are not associated with any specific Sentential sentence, but with arbitrarily chosen sentences which have the truth value in question. Whatever Sentential sentence we choose in place of $\mathcal{A}$, whether atomic or not, we know that if $\mathcal{A}$ has the truth value True, then $\neg\mathcal{A}$ will have the truth value False.

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. We can summarize this in the schematic 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 COMMUTATIVE. 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. We can summarize this in the schematic 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 commutative.

Conditional I'm just going to come clean and admit it. Conditionals are a problem in Sentential. This is not because there is any problem finding a truth table for the connective '→', but rather because the truth table we put forward seems to make '→' behave in ways that are different to the way that the English counterpart 'if ... then ...' behaves. Exactly how much of a problem this poses is a matter of *philosophical* contention. I shall discuss a few of the subtleties in §8.6 and §11.5. (It is no problem for Sentential itself, of course – the only potential difficulty arises when we try to use the Sentential conditional to represent 'if ..., then ....')

We know at least this much from a parallel with the English conditional: if $\mathcal{A}$ is true and $\mathcal{B}$ is false, then $\mathcal{A} \to \mathcal{B}$ should be false. The conditional claim 'if I study hard, then I'll pass' is clearly false if you study hard and still fail. For now, I am going to stipulate that this is the only type of case in which $\mathcal{A} \to \mathcal{B}$ is false. We can summarize this with a schematic truth table for the Sentential conditional.

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

The conditional is *not* commutative. You cannot swap the antecedent and consequent in general without changing the truth value, because $\mathcal{A} \to \mathcal{B}$ has a different truth table from $\mathcal{B} \to \mathcal{A}$. Compare:

1. If a coin lands heads 1000 times, something surprising has happened. (True – that is very surprising.)
2. If something surprising has happened, then a coin lands heads 1000 times. (False – there are other surprising things than that.)

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

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

You can think of the biconditional as saying that the two immediate constituents have the same truth value – it's true if they do, false otherwise. Unsurprisingly, the biconditional is commutative.

8.4 Symbolising Versus Translating

We have seen how to use a symbolisation key in §4.1 to *temporarily* assign an interpretation to some of the atomic sentences of Sentential. This symbolisation key will at the very least assign a truth-value to those atomic sentences – the same one as the English sentence it symbolises.

But since the connectives of Sentential are truth-functional, they really are sensitive to nothing *except* the truth values of the symbolised sentences. So when we are symbolising a sentence or an argument in Sentential, we are ignoring everything *besides* the contribution that the truth values of a subsentence might make to the truth value of the whole.

There are subtleties to natural language sentences 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 Sentential. As remarked in §5, Sentential cannot capture the subtle differences between the following English sentences:

1. Jon is elegant and Jon is quick
2. Although Jon is elegant, Jon is quick
3. Despite being elegant, Jon is quick
4. Jon is quick, albeit elegant
5. Jon's elegance notwithstanding, he is quick

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

I keep saying that we use Sentential sentences to *symbolise* English sentences. Many other textbooks talk about *translating* English sentences into Sentential. But a good translation should preserve certain facets of meaning, and – as I have just pointed out – Sentential just cannot do that. This is why I 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: Jon is elegant.
Q: Jon is quick.

Other textbooks will understand this as a stipulation that the Sentential sentence 'F' should *mean* that Jon is elegant, and that the Sentential sentence 'Q' should *mean* that Jon is quick. But Sentential just is totally unequipped to deal with *meaning*. The preceding symbolisation key is doing no more nor less than stipulating that the Sentential sentence 'F' should take the same truth value as the English sentence 'Jon is elegant' (whatever that might be), and that the Sentential sentence 'Q' should take the same truth value as the English sentence 'Jon is quick' (whatever that might be).

When we treat an atomic Sentential sentence as *symbolising* an English sentence, we are stipulating that the Sentential sentence is to take the same truth value as that English sentence.
When we treat a compound Sentential sentence as symbolising an English sentence, we are claiming that they share a truth-functional structure, and that the atomic sentences of the Sentential sentence symbolise those sentences which play a corresponding role in the structure of the English sentence.

8.5 Non-Truth-Functional Connectives

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

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

Whereas it is necessarily the case that $2 + 2 = 4$,[1] it is not *necessarily* the case that Shostakovich wrote fifteen string quartets. If Shostakovich had died earlier, he would have failed to finish Quartet no. 15; if he had lived longer, he might have written a few more. So 'It is necessarily the case that ...' is a connective of English, but it is not *truth-functional*. Another example: 'one hundred years ago'. Both 'many people have cars' and 'many people have children' are true. But while 'One hundred years ago, many people had children' is true, 'One hundred years ago, many people had cars' is false.

In these cases, we had the same input truth values, but different output. We can turn this into a *test* for truth-functionality: if for some one-place connective '#', you can find sentences $\mathcal{A}$ and $\mathcal{B}$ such that (i) $\mathcal{A}$ has the same truth value as $\mathcal{B}$, while (ii) $\#\mathcal{A}$ has a different truth value from $\#\mathcal{B}$, then '#' is not a truth-functional connective. Its truth value is obviously not fixed by the truth value of its immediate subsentence. The test can be generalised in the obvious way to binary connectives, etc: can you find a connective such that when you feed it the same truth values as input, you get different results? If so, that result is not a function of the input truth values.

8.6 Indicative Versus Subjunctive Conditionals

I want to bring home the point that Sentential *only* deals with truth functions by considering the case of the conditional. When I introduced the schematic truth table for the material conditional in §8.3, I did not say much to justify it. Let me now offer two justifications. These are not arguments for the truth function associated with '→', which is

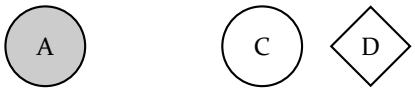
[1] Given that the English numeral '2' names the number two, and the numeral '4' names the number four, and '+' names addition, then it must be that the result of adding two to itself is four. This is not to say that '2' had to be used in the way we actually use it – if '2' had named the number three, the sentence would have been false. But in its actual use, it is a necessary truth.

a perfectly legitimate rule for associating input truth values with outputs. What needs justification is our idea that we should use this truth function when symbolise English sentences whose main connective is 'if ... then ...'. They attempt to show that our usage of the English conditional 'if' is in line with the way '→' behaves.

Edgington's Argument The first follows a line of argument due to Dorothy Edgington.[2] Suppose that Lara has drawn some shapes on a piece of paper, and coloured some of them in. I have not seen them, but I claim:

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

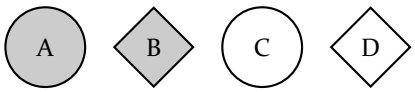
As it happens, Lara has drawn the following:



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

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

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



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

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

Now, recall that every connective of Sentential 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.

[2] Dorothy Edgington (2020) 'Indicative Conditionals', in Edward N Zalta, ec., *TheStanford Encyclopedia of Philosophy*, http://plato.stanford.edu/entries/conditionals/.

The or-to-if argument A second justification for symbolising 'if' as '→' is this. We know already that if $\mathcal{A}$ is true and $\mathcal{C}$ is false, then 'if $\mathcal{A}$ then $\mathcal{C}$' will be false. What should we say about the other rows of the truth table, the rows on which either $\mathcal{A}$ is false, or $\mathcal{B}$ is true (or both)? On these lines either $\mathcal{A}$ is false or $\mathcal{B}$ is true. So the disjunction 'Not-$\mathcal{A}$ or $\mathcal{B}$' is true on these lines. If a disjunction is true, and its first disjunct isn't, then the second disjunct has to be true. In this case, the first disjunct ('Not-$\mathcal{A}$') isn't true just in case $\mathcal{A}$ is true; so it turns out that if $\mathcal{A}$ obtains, then so does $\mathcal{B}$. So the truth of the disjunction on those three lines leads us to conclude that the conditional 'if $\mathcal{A}$, then $\mathcal{B}$' should also be true on those three lines. Thus we obtain the truth table we have associated with '→' as the best truth table to use for symbolising 'if'.

What these two arguments show is that '→' is the *only* candidate for a truth-functional conditional. Otherwise put, *it is the best conditional that Sentential can provide*. But is it any good, as a surrogate for the conditionals we use in everyday language? Should we think that 'if' is a truth functional connective? Consider two sentences:

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

Sentence 65 is true; sentence 66 is false. But both have false antecedents and false consequents. So the truth value of the whole sentence is not uniquely determined by the truth value of the parts. This use of 'if' fails our test for truth-functionality. Do not just blithely assume that you can adequately symbolise an English 'if ..., then ...' with Sentential's '→'.

The crucial point is that sentences 65 and 66 employ SUBJUNCTIVE conditionals, rather than INDICATIVE conditionals. Subjunctive conditionals are also sometimes known as COUNTERFACTUALS. They ask us to imagine something contrary to what we are assuming as fact – that Mitt Romney lost the 2012 election – and then ask us to evaluate what *would* have happened in that case. The classic illustration of the difference between the indicative and subjunctive conditional comes from pairs like these:

67. If a dingo *didn't take* Azaria Chamberlain, something else *did*.
68. If a dingo *hadn't taken* Azaria Chamberlain, something else *would have*.

The indicative conditional in 67 is true, given the actual historical fact that she was taken, and given that we are not assuming at this point anything about how she was taken. But is the subjunctive in 68 also true? It seems not. She was not destined to be taken by something or other, and if the dingo hadn't intervened, she wouldn't have disappeared at all.[3]

[3] If we are assuming as fact that a dingo took her, then when we consider what would have happened had the dingo not been involved, we imagine a situation in which all the actual consequences of the dingo's action are removed.

The point to take away from this is that subjunctive conditionals cannot be tackled using '→'. This is not to say that they cannot be tackled by any formal logical language, only that Sentential is not up to the job.[4]

So the '→' connective of Sentential is at best able to model the indicative conditional of English, as in 67. In fact there remain difficulties even with indicatives in Sentential. One family of difficulties arises from consideration of the or-to-if argument. The argument seems compelling in cases like this:

69. Either the butler or the gardener did it;
So: If it wasn't the butler, it was the gardener.

But what if our confidence in the premise 69 derives from our confidence in just one disjunct? Suppose that we are certain it was the butler, and certain that the gardener has an airtight alibi and wasn't anywhere near the manor at the time of the murder? Because we are certain it was the butler, we might be equally certain of 69, that it was either the butler or the gardener (this inference from a disjunct to a disjunction seems odd, but it is surely valid). But we might also be sure that if it wasn't the butler, it was the valet – he was the only other person with motive. In this sort of case, we might have confidence in the premise 69 of this or-to-if argument and reject its conclusion. Yet the Sentential analogue of the or-to-if argument is valid: as we'll see after we introduce the concept of validity for Sentential in §11), '$A \vee B \therefore \neg A \rightarrow B$' turns out to be valid. This mismatch suggests that 'if' and '→' aren't a perfect match. I shall say a little more about other difficulties for the material conditional analysis of indicatives in §11.5 and in §30.1.

For now, I shall content myself with the observation that '→' is the only plausible candidate for a truth-functional conditional. Our working hypothesis is that many uses of 'if' can be *adequately approximated* by '→'. Many English conditionals cannot be represented adequately using '→'. Sentential is an intrinsically limited language. But this is only a problem if you try to use it to do things it wasn't designed to do.

Key Ideas in §8

- The connectives of Sentential are all truth-functional, and have their meanings specified by the truth-tables laid out in §8.3.
- When we treat a sentence of Sentential as symbolising an English sentence, we need only say that as far as truth value is concerned and truth-functional structure is concerned, they are alike.
- English has many nontruth-functional connectives. Some uses of the conditional 'if' are nontruth-functional. But as long as we remain aware of the limitations of Sentential, it can be a very powerful tool for modelling a significant class of arguments.

4 There are in fact logical treatments of counterfactuals, the most influential of which is David Lewis (1973) *Counterfactuals*, Blackwell.

Practice exercises

A. Which of the following arguably may *not* characterise a function, where x is the input and y is the output:

1. y is the product of x and itself;
2. y is the square root of x;
3. y is a child of x;
4. y is a child of x and younger than any other child of x;
5. y is taller than x;
6. y is as tall as x.

B. *True or false*: if the main connective of some sentence is truth-functional, then the truth value of the sentence uniquely determines the truth values of any constituents.

C. Suppose † is some English one-place connective, so that '†$\mathcal{A}$' is a grammatical sentence. How can we test if it is not truth-functional?

9

Complete Truth Tables

9.1 Valuations

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

> A VALUATION is any assignment of truth values to some atomic sentences of Sentential. It assigns exactly one truth value, either True or False, to each of the sentences in question.

A valuation is thus a function from atomic sentences to truth values. So this is a valuation:

$$A, G, P, G_7 \mapsto \mathrm{T}$$
$$F, R, Z \mapsto \mathrm{F}.$$

There is no requirement that a valuation be a TOTAL FUNCTION, that is, that it assign a truth value to every atomic sentence. To be fix the truth value of a sentence $\mathcal{A}$ of Sentential a valuation must assign a truth value to every atomic sentence $\mathcal{A}$ contains.

A valuation is a temporary assignment of 'meanings' to Sentential sentences, in much the same way as a symbolisation key might be. (It only assigns truth values, the only dimension of meaning that Sentential is sensitive to.) What is distinctive about Sentential is that almost all of its basic vocabulary – the atomic sentences – only get their meanings in this temporary fashion. The only parts of Sentential that get their meanings permanently are the connectives, which always have a fixed interpretation.

This is rather unlike English, where most words have their meanings on a permanent basis. But there are some words in English – like pronouns ('he', 'she', 'it') and demonstratives ('this', 'that') – that get their meaning assigned temporarily, and then can

be reused with a different meaning in another context. Such expressions are called CONTEXT SENSITIVE. In this sense, all the atomic sentences of Sentential are context sensitive expressions. Of course we don't have anything so explicit and deliberate as a valuation or a symbolisation key in English to assign a meaning to a particular use of 'this' or 'that' – the circumstances of a conversation automatically assign an appropriate object (usually). In Sentential, however, we need to explicitly set out the interpretations of the atomic sentences we are concerned with.

9.2 Truth Tables

We introduced schematic truth tables in §8.3. These showed what truth value a compound sentence with a certain structure was determined to have by the truth values of its subsentences, whatever they might be. We now introduce a closely related idea, that of a truth table. This shows how a specific compound sentence has its truth value determined by the truth values of its specific *atomic subsentences*, across all the possible ways that those atomic subsentences might be assigned True and False.

You will no doubt have realised that a way of assigning True and False to atomic sentences is a valuation. So we can say: a TRUTH TABLE summarises how the truth value of a compound sentence depends on the possible valuations of its atomic subsentences. Each row of a truth table represents a possible valuation. The entire complete 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 pretty abstract. So it might be easiest to explain with an example.

9.3 A Worked Example

Consider the sentence '$(H \wedge I) \rightarrow H$'. There are four possible ways to assign True and False to the atomic sentences 'H' and 'I': both true, both false, 'H' true and 'I' false, and 'I' true and 'H' false. So there are four possible valuations of these two atomic sentences. We can lay out these valuations as follows:

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

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

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

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

		$\mathcal{A} \wedge \mathcal{B}$			
H	I	$(H$	$\wedge I)$	$\rightarrow$	H
T	T	T	T	T	T
T	F	T	F	F	T
F	T	F	F	T	F
F	F	F	F	F	F

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

		$\mathcal{C}$		$\rightarrow \mathcal{D}$
H	I	$(H$	$\wedge I)$	$\rightarrow H$
T	T	T	T	T
T	F	F	T	T
F	T	F	T	F
F	F	F	T	F

The conditional is the main logical connective of the sentence. And the column of 'T's underneath the conditional tells us that the sentence '$(H \wedge I) \rightarrow H$' is true regardless of the truth values of 'H' and 'I'. They can be true or false in any combination, and the compound sentence still comes out true. Since we have considered all four possible assignments of truth and falsity to 'H' and 'I' – since, that is, we have considered all the different *valuations* – we can say that '$(H \wedge I) \rightarrow H$' is true on every valuation.

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

H	I	$(H$	$\wedge I)$	$\rightarrow$	H
T	T	T	T	**T**	T
T	F	T	F	**T**	T
F	T	F	F	**T**	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 connective* for the sentence, since this tells you the truth value of the entire sentence. I have emphasised this, by putting this column in bold. When you work through truth tables yourself, you should similarly emphasise it (perhaps by drawing a box around the relevant column).

9.4 Building Complete Truth Tables

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

The size of the complete truth table depends on the number of different atomic sentences in the table. A sentence that contains only one atomic sentence requires only two rows, as in the schematic truth table for negation. This is true even if the same letter is repeated many times, as in the sentence '$\big((C \leftrightarrow C) \rightarrow C\big) \wedge \neg(C \rightarrow C)$'. The complete truth table requires only two rows 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	$\big((C$	$\leftrightarrow$	$C)$	$\rightarrow$	$C\big)$	$\wedge$	$\neg$	$(C$	$\rightarrow$	$C)$
T	T	T	T	T	T	**F**	F	T	T	T
F	F	T	F	F	F	**F**	F	F	T	F

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

A sentence that contains two atomic sentences requires four rows for a complete truth table, as in the schematic truth tables, and as in the complete truth table for '$(H \wedge I) \rightarrow H$'.

A sentence that contains three atomic sentences requires eight rows:

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

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

A complete truth table for a sentence that contains four different atomic sentences requires 16 rows. Five letters, 32 rows. Six letters, 64 rows. And so on. To be perfectly general: If a complete truth table has n different atomic sentences, then it must have 2^n rows.[1]

In order to fill in the columns of a complete 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 row truth table like the one above. For a 16 row truth table, the next column of atomic sentences should have eight 'T's followed by eight 'F's. For a 32 row table, the next column would have 16 'T's followed by 16 'F's. And so on.

Key Ideas in §9

- A valuation of some atomic sentences associates each of them with exactly one of our truth values; it is like an extremely stripped down version of a symbolisation key. In there are n atomic sentences, there are 2^n valuations of them.
- A truth table lays out the truth values of a particular Sentential sentence in each of the distinct possible valuations of its constituent atomic sentences.

Practice exercises

A. How does a *schematic* truth table differ from a regular truth table? What is a *complete* truth table?

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

1. $A \to A$
2. $C \to \neg C$
3. $(A \leftrightarrow B) \leftrightarrow \neg(A \leftrightarrow \neg B)$
4. $(A \to B) \lor (B \to A)$
5. $(A \land B) \to (B \lor A)$
6. $\neg(A \lor B) \leftrightarrow (\neg A \land \neg B)$
7. $\big((A \land B) \land \neg(A \land B)\big) \land C$
8. $\big((A \land B) \land C\big) \to B$
9. $\neg\big((C \lor A) \lor B\big)$

If you want additional practice, you can construct truth tables for any of the sentences and arguments in the exercises for Chapter 2.

[1] Since the values of atomic sentences are independent of each other, each new atomic sentence $\mathcal{A}_{n+1}$ we consider is capable of being true or false on every existing valuation on $\mathcal{A}_1, \ldots, \mathcal{A}_n$, and so there must be twice as many valuations on $\mathcal{A}_1, \ldots, \mathcal{A}_n, \mathcal{A}_{n+1}$ as on $\mathcal{A}_1, \ldots, \mathcal{A}_n$.

10

Semantic Concepts

In §9.1, we introduced the idea of a valuation and showed how to determine the truth value of any Sentential sentence on any valuation using a truth table in the remainder of the chapter. In this section, we shall introduce some related ideas, and show how to use truth tables to test whether or not they apply.

10.1 Logical Truths and Falsehoods

In §3, I explained *necessary truth* and *necessary falsity*. Both notions have close but imperfect surrogates in Sentential. We shall start with a surrogate for necessary truth.

> $\mathcal{A}$ is a LOGICAL TRUTH iff it is true on every valuation (among those valuations on which it has a truth value).

We need the parenthetical clause because of the way we have defined valuations. A given valuation v might only assign truth values to some atomic sentences and not all. For any sentence $\mathcal{A}$ which contains an atomic sentence to which v doesn't assign a truth value, $\mathcal{A}$ will not have any truth value according to v. Logical truths in Sentential are sometimes called TAUTOLOGIES.

We can determine whether a sentence is a logical truth just by using truth tables. If the sentence is true on every row of a complete truth table, then it is true on every valuation for its constituent atomic sentences, so it is a logical truth. In the example of §9, '$(H \wedge I) \rightarrow H$' is a logical truth.

This is only, though, a surrogate for necessary truth. There are some necessary truths that we cannot adequately symbolise in Sentential. An example is '$2 + 2 = 4$'. This *must* be true, but if we try to symbolise it in Sentential, the best we can offer is an

atomic sentence, and no atomic sentence is a logical truth.[1] Still, if we can adequately symbolise some English sentence using a Sentential sentence which is a logical truth, then that English sentence expresses a necessary truth.

We have a similar surrogate for necessary falsity:

> $\mathcal{A}$ is a LOGICAL FALSEHOOD iff it is false on every valuation (among those on which it has a truth value).

We can determine whether a sentence is a logical falsehood just by using truth tables. If the sentence is false on every row of a complete truth table, then it is false on every valuation, so it is a logical falsehood. In the example of §9, '$\big((C \leftrightarrow C) \rightarrow C\big) \wedge \neg(C \rightarrow C)$' is a logical falsehood. A logical falsehood is often called a CONTRADICTION.

10.2 Logical Equivalence

Here is a similar, useful notion:

> $\mathcal{A}$ and $\mathcal{B}$ are LOGICALLY EQUIVALENT iff they have the same truth value on every valuation among those which assign both of them a truth value.

It is easy to test for logical equivalence using truth tables. Consider the sentences '$\neg(P \vee Q)$' and '$\neg P \wedge \neg Q$'. Are they logically equivalent? To find out, we may construct a truth table.

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

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

[1] At this risk of repeating myself: $2 + 2 = 4$ is necessarily true, but it is not necessarily true *in virtue of its structure*. A necessary truth is true, *with its actual meaning*, in every possible situation. A Sentential-logical truth is true *in the actual situation* on every possible way of interpreting its atomic sentences. These are interestingly different notions.

10.3 More Parenthetical Conventions

Consider these two sentences:

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

These have the same truth table, and are logically equivalent. Consequently, it will never make any difference from the perspective of truth value – which is all that Sentential cares about (see §8) – which of the two sentences we assert (or deny). And since the order of the parentheses does not matter, I shall allow us 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 parentheses. (I already allowed us to drop outermost parentheses in §6.) The same observation holds for disjunctions. Since the following sentences are logically equivalent:

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

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 parentheses. *But be careful*. These two sentences have *different* truth tables, so are not logically equivalent:

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

So if we were to write:

$$A \rightarrow B \rightarrow C$$

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

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

So if we were to write:

$$A \vee B \wedge C$$

it would be dangerously ambiguous. *Never write this.* The moral is: you can drop parentheses when dealing with a long list of conjunctions, or when dealing with a long list of disjunctions. But that's it.

10.4 Consistency

In §3, I 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 CONSISTENT iff there is some valuation which makes them all true.

Derivatively, sentences are JOINTLY INCONSISTENT if there is no valuation that makes them all true. Note that this notion applies to a single sentence as well: a sentence is consistent iff it is true on some valuation.

Again, it is easy to test for joint consistency using truth tables. If we draw up a truth table for all the sentences together, if there is some row on which each of them gets a 'T', then they are consistent.

So, for example, consider these sentences: $\neg P$, $P \to Q$, Q:

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

We can see on the third row, the valuation which assigns F to 'P' and T to 'Q', each of the sentences is true. So these are jointly consistent.

> **Key Ideas in §10**
>
> - A logical truth is a sentence true on every valuation of its atomic constituents; a logical falsehood is true on no valuation of its atomic constituents.
> - A sentence (collection of sentences) is consistent iff there is a valuation on which it is (they are all) true.
> - Two sentences are logically equivalent iff they are true on exactly the same valuations. We can use the notion of logical equivalence to motivate further parenthetical conventions.

Practice exercises

A. 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.

B. What is the difference between a logical truth and a logical falsehood? Are there any other kinds of sentence in Sentential?

Revisit your answers to exercise §9B (page 79). Determine which sentences were logical truths, which were logical falsehoods, and which, if any, were neither logical truths nor logical falsehoods.

C. What does it mean to say that two sentences of Sentential are *logically equivalent*? Use truth tables to decide if the following pairs of sentences are logically equivalent:

1. $\neg(P \wedge Q), (\neg P \vee \neg Q)$;
2. $(P \rightarrow Q), \neg(Q \rightarrow P)$;
3. $\neg(P \leftrightarrow Q), ((P \vee Q) \wedge \neg(P \wedge Q))$.

D. What does it mean to say that some sentences of Sentential are jointly inconsistent?

Use truth tables to determine whether these sentences are jointly consistent, or jointly inconsistent:

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

11

Entailment and Validity

11.1 Entailment

The following idea is related to joint consistency, but is of great interest in its own right:

> The sentences $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ ENTAIL the sentence $\mathcal{C}$ if 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.

(Why is this not a biconditional? The full answer will have to wait until §23.)

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

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

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

11.2 The Double Turnstile

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

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

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

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

70. $P, P \to Q \vDash Q$

is just an abbreviation for the English sentence:

71. The Sentential sentences 'P' and '$P \to Q$' entail 'Q'.

Note that there are no constraints on the number of Sentential sentences that can be mentioned before the symbol '$\vDash$'. Indeed, one limiting case is of special interest:

72. $\vDash \mathcal{C}$.

72 is false if there is a valuation which makes all the sentences appearing on the left hand side of '$\vDash$' true and makes $\mathcal{C}$ false. Since *no* sentences appear on the left side of '$\vDash$' in 72, it is trivial to make 'them' all true. So it is false if there is a valuation which makes $\mathcal{C}$ false – and so 72 is true iff every valuation makes $\mathcal{C}$ true. Otherwise put, 72 says that $\mathcal{C}$ is a logical truth. Equally:

73. $\mathcal{A}_1, \ldots, \mathcal{A}_n \vDash$

says that $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are jointly inconsistent. It follows that

74. $\mathcal{A} \vDash$

says that $\mathcal{A}$ is individually inconsistent. That is to say, $\mathcal{A}$ is a logical falsehood.[1]

Here is the important connection between inconsistency and entailment, expressed using this new notation.

$$\boxed{\mathcal{A}_1, \ldots, \mathcal{A}_n \vDash \mathcal{C} \text{ iff } \mathcal{A}_1, \ldots, \mathcal{A}_n, \neg\mathcal{C} \vDash.}$$

If every valuation which makes each of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ true also makes $\mathcal{C}$ true, then all of those valuations also make '$\neg\mathcal{C}$' false. So there can be no valuation which makes each of $\mathcal{A}_1, \ldots, \mathcal{A}_n, \neg\mathcal{C}$ true. So those sentences are jointly inconsistent.

[1] If you find it difficult to see why '$\vDash \mathcal{A}$' should say that $\mathcal{A}$ is a logical truth, you should just take 72 as an abbreviation for that claim. Likewise you should take '$\mathcal{A} \vDash$' as abbreviating the claim that $\mathcal{A}$ is a logical falsehood.

11.3 '⊨' versus '→'

I now want to compare and contrast '⊨' and '→'.

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

Observe: $\mathcal{A} \to \mathcal{C}$ is a logical truth 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, $\mathcal{A} \to \mathcal{C}$ is a logical truth iff there is no valuation that makes $\mathcal{A}$ true and $\mathcal{C}$ false.

Combining these two observations, we see that $\mathcal{A} \to \mathcal{C}$ is a logical truth iff $\mathcal{A} \vDash \mathcal{C}$.[2] But there is a really important difference between '⊨' and '→':

> '→' is a sentential connective of Sentential.
> '⊨' is a symbol of augmented English.

When '→' is flanked with two Sentential sentences, the result is a longer Sentential sentence. By contrast, when we use '⊨', we form a metalinguistic sentence that *mentions* the surrounding Sentential sentences.

If $\mathcal{A} \to \mathcal{C}$ is a logical truth, then $\mathcal{A} \vDash \mathcal{C}$. But $\mathcal{A} \to \mathcal{C}$ can be true on a valuation without being a logical truth, and so can be true on a valuation even when $\mathcal{A}$ doesn't entail $\mathcal{C}$. Sometimes people are inclined to confuse entailment and conditionals, perhaps because they are tempted by the thought that we can only establish the truth of a conditional by *logically deriving* the consequent from the antecedent. But while this is the way to establish the truth of a logically true conditional, most conditionals posit a weaker relation between antecedent and consequent than that – for example, a causal or statistical relationship might be enough to justify the truth of the conditional 'If you smoke, then you'll lower your life expectancy'.

11.4 Entailment and Validity

We now make an important observation:

> If $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{C}$, then $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ is valid.

Here's why. If $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ entail $\mathcal{C}$, then there is no valuation which makes all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ true whilst making $\mathcal{C}$ false. It is thus not possible – given the actual meanings of the connectives of Sentential – for the sentences $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ to jointly be true without $\mathcal{C}$ being true too, so the argument $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ is conclusive. Furthermore, because the conclusiveness of this argument doesn't depend on anything other that the structure of the sentences in the argument, it is also valid.

[2] This result sometimes goes under a fancy title: *the Deduction Theorem for Sentential*. It is easy to see that this more general result follows from the Deduction Theorem: $\mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{A} \vDash \mathcal{C}$ iff $\mathcal{B}_1, \ldots, \mathcal{B}_n, \vDash \mathcal{A} \to \mathcal{C}$.

The only conclusive arguments in Sentential are valid ones. Because we consider every valuation, the conclusiveness of an argument cannot turn on the particular truth values assigned to the atomic sentences. For any collection of atomic sentences, there is a valuation corresponding to any way of assigning them truth values. This means that we treat the atomic sentences as all independent of one another. So there is no possibility that there might be some connection in meaning between sentences of Sentential *unless* it is in virtue of those sentences having shared constituents and the right structure.

In short, we have a way to test for the validity of some English arguments. First, we symbolise them in Sentential, as having premises $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$, and conclusion $\mathcal{C}$. Then we test for entailment using truth tables. If there is an entailment, then we can conclude that the argument we symbolised has the right kind of structure to count as valid.

For example, suppose we consider this argument:

> Jim studied hard, so he didn't act in a lot of plays. For he can't study hard while acting in lots of plays.

The argument can be paraphrased as follows, a form we saw in §4.1:

> It's not the case that Jim both studied hard and acted in lots of plays.
> Jim studied hard
> So: Jim did not act in lots of plays.

We offer this symbolisation key:

> S: Jim studied hard;
> A: Jim acted in lots of plays.

Using this symbolisation, the argument has the following form:

$$\neg(S \wedge A), S \therefore \neg A.$$

We draw up a truth table:

A	S	$\neg(S \wedge A)$	S	$\neg A$
T	T	F	T	F
T	F	T	F	F
F	**T**	**T**	**T**	**T**
F	F	T	F	T

The only valuation on which the premises are both true is represented on the third row, assigning F to A and T to S. And on this valuation, the conclusion is true. So this argument is valid, and this statement of entailment is correct: $\neg(S \wedge A), S \vDash \neg A$.

Our test uses the precisely defined relation of entailment in Sentential as a test for validity in natural language. If we have symbolised an argument successfully, then we have captured its form (or rather, one of its forms). If the symbolised Sentential argument turns out to be valid, we can conclude that the natural language argument is valid too, because it can be modelled as having a valid form.

11.5 The Limits of these Tests

We have reached an important milestone: a test for the validity of arguments! But, we should not get carried away just yet. It is important to understand the *limits* of our achievement. There are three sorts of limitations I want to discuss:

1. Some valid arguments are overlooked by our test;
2. Some invalid arguments are misclassified by a naive application of our test to an inadequate symbolisation; and
3. There are some putative examples of sentences that cannot be symbolised because of some assumptions that Sentential makes.

I shall illustrate these limits with three examples.

First, consider the argument:

75. Daisy is a small cow. So, Daisy is a cow.

To symbolise this argument in Sentential, we would have to use two different atomic sentences – perhaps 'S' and 'C' – for the premise and the conclusion respectively. Now, it is obvious that 'S' does not entail 'C'. But the English argument surely seems valid – the structure of 'Daisy is a small cow' guarantees that 'Daisy is a cow' is true. Note that a small cow might still be rather large, so we cannot fudge things by symbolising 'Daisy is a small cow' as a conjunction of 'Daisy is small' and 'Daisy is a cow'. (We'll return to this sort of case in §16, where we will see how to symbolise 75 as a valid argument in Quantifier.) But our Sentential-based test for validity in English will have some *false negatives*: it will classify some valid English arguments as invalid. This is because some valid arguments are valid in virtue of structure which is not truth-functional. We'll see more examples of this in §15.

Second, consider the following arguments:

76. It's not the case that the Crows will win by a lot, if they win. So the Crows will win.
77. It's not the case that, if God exists, She answers malevolent prayers. So God exists.

Both of these arguments have the same structure. Let's focus on the second, example 77. Symbolising it in Sentential, we would offer something like '$\neg(G \rightarrow M) \therefore G$'. Now, as can easily be checked with a truth table, this is a correct entailment in Sentential. So if we symbolise the argument 77 in Sententialin this way, the conditional premise entails that God exists. But that's strange: surely even the atheist can accept sentence 77, without contradicting herself! Some say that 77 would be better symbolised by '$G \rightarrow \neg M$', even though that doesn't reflect the apparent form of the English sentence. '$G \rightarrow \neg M$' does not entail G. This symbolisation does a better job of reflecting the intuitive consequences of the English sentence 77, but at the cost of abandoning a

straightforward correspondence between the structure of English sentences and their Sentential symbolisations.

A better alternative might be to think that the conditional 'if God exists, she answers malevolent prayers' is not to be symbolised by '→'. This conditional is false, many think: they may think, for example, that it is part of the concept of God that God is good, and hence would not grant prayers with evil intent. Even the atheist might accept this, maybe because they accept the subjunctive conditional 'even if God were to exist, God would not answer malevolent prayers' (§8.6). This sort of example has motivated many philosophers to offer nontruth-functional accounts of the English 'if', including some that make it behave rather like a subjunctive conditional.[3]

The cases in 76 and 77 are examples of the (so-called) *paradoxes of material implication*. They highlight a limitation of Sentential, in that it appears not to have a conditional connective that adequately models these uses of the English 'if'. But it is also a limitation of our tests for validity in English, because the test is only as good as the symbolisations we come up with as part of it. If we are not careful, we might end up mistakenly using an inadequate symbolisation, and giving the wrong verdict about some argument, thinking it valid when it is not. (I will return one final time to the relation between 'if' and → in §30.1.)

Finally, consider the sentence:

78. Jan is neither bald nor not-bald.

To symbolise this sentence in Sentential, we would offer something like '$\neg(J \vee \neg J)$'. This a logical falsehood (check this with a truth-table). But sentence 78 does not itself seem like a logical falsehood; for we might have happily go on to add 'Jan is on the borderline of baldness'! To make this point another way: as is easily seen by truth tables, '$\neg(J \vee \neg J)$' is logically equivalent to '$\neg J \wedge J$'. This latter sentence symbolises an obvious logical falsehood in English:

79. Jan is both not-bald and also bald.

It is so obvious, though, that 78 is synonymous with 79? It seems like it may not be, even though our test will classify any English argument from one to the other as valid (since both are symbolised as logical falsehoods, which degenerately entail anything).

Because of the way we have defined valuations, *every* sentence of Sentential is assigned either True or False in any valuation which makes it meaningful by assigning truth values to its atomic sentences. This property of Sentential is known as BIVALENCE: that every sentence has exactly one of the two possible truth values. The case of Jan's baldness (or otherwise) raises the general question of what logic we should use when dealing with *vague* discourse, properties like 'bald' or 'tall' which seem to have borderline cases. Many think it plausible that a borderline case of F is neither a case of F, nor does

[3] Edgington discusses Stalnaker's version of such an account in 'Nearest Possible Worlds', §4.1 of her 'Indicative Conditionals', cited above (http://plato.stanford.edu/entries/conditionals/#Sta).

it fail to be a case of F. Hence they have been tempted to deny bivalence for English: 'Jan is bald', they say, is neither True nor False! If *p* is neither true nor false, then it is hardly surprising that '*p* or not-*p*' turns out to be untrue. If these thinkers are right that vagueness in English leads to the denial of bivalence, while Sentential is bivalent, this will give rise to mismatches between English and Sentential. These mismatches will not involve inadequate symbolisation, but a more fundamental disagreement about the background framework – here, a disagreement about the nature of truth.[4]

In different ways, these three examples highlight some of the limits of working with a language like Sententialthat can *only* handle truth-functional connectives. Moreover, these limits give rise to some interesting questions in philosophical logic. Part of the purpose of this course is to equip you with the tools to explore these questions of *philosophical logic*. But we have to walk before we can run; we have to become proficient in using Sentential, before we can adequately discuss its limits, and consider alternatives. It is important to recognise that these are limits to Sentential only in its role as a framework to model validity in English and other natural languages. They are not problems for Sentential as a formal language. Moreover, as I have emphasised already, these limitations are merely manifestations of the fact that Sentential is being used as a model of natural language. Models are typically not designed or intended to capture every aspect of what they model. Their utility derives often from being simpler than the complex things they are representing. The limitations we have noted indicate that Sentential may not model English perfectly in these cases. But Sentential remains an adequate model of English in many other cases.

Key Ideas in §11

- If every valuation which makes some sentences all true is also one that makes some further sentence true, then those sentences entail the further sentence. We use the symbol '⊨' for entailment.
- We can test for entailment using truth tables, in the same sort of way that we test for consistency.
- If an argument when symbolised turns out to be an entailment, then the original argument is valid in virtue of its truth-functional structure. So we can test for validity using the truth table tests for entailment.
- These tests nevertheless have limitations: not every valid argument can be symbolised as a Sentential entailment. These limitations are typical of using simpler models to represent complex things.

4 For more on the logic of vagueness, see Roy Sorensen's 2018 entry 'Vagueness' in *The Stanford Encyclopedia of Philosophy* (https://plato.stanford.edu/entries/vagueness/). He discusses views that deny bivalence in §5.

Practice exercises

A. What does it mean to say that sentences $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ of Sentential entail a further sentence $\mathcal{C}$?

B. If $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{C}$, what can you say about the argument with premises $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ and conclusion $\mathcal{C}$?

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

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

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

1. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are logically equivalent. What can you say about $\mathcal{A} \leftrightarrow \mathcal{B}$?
2. Suppose that $(\mathcal{A} \land \mathcal{B}) \to \mathcal{C}$ is neither a logical truth nor a logical falsehood. What can you say about whether $\mathcal{A}, \mathcal{B} \therefore \mathcal{C}$ is valid?
3. Suppose that $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are jointly inconsistent. What can you say about $(\mathcal{A} \land \mathcal{B} \land \mathcal{C})$?
4. Suppose that $\mathcal{A}$ is a logical falsehood. What can you say about whether $\mathcal{A}, \mathcal{B} \vDash \mathcal{C}$?
5. Suppose that $\mathcal{C}$ is a logical truth. What can you say about whether $\mathcal{A}, \mathcal{B} \vDash \mathcal{C}$?
6. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are logically equivalent. What can you say about $(\mathcal{A} \lor \mathcal{B})$?
7. Suppose that $\mathcal{A}$ and $\mathcal{B}$ are *not* logically equivalent. What can you say about $(\mathcal{A} \lor \mathcal{B})$?

E. If two sentences of Sentential, $\mathcal{A}$ and $\mathcal{D}$, are logically equivalent, what can you say about $(\mathcal{A} \to \mathcal{D})$? What about the argument $\mathcal{A} \therefore \mathcal{D}$?

F. Consider the following principle:

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

Is this principle correct? Explain your answer.

12

Truth Table Shortcuts

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

12.1 Working through Truth Tables

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

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

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

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

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

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

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

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

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

12.2 Testing for Validity and Entailment

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

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

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

With this in mind, consider how we might test the following claimed entailment:

$$\neg L \to (J \vee L), \neg L \vDash J.$$

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

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

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

The easiest premise on the left of the turnstile to evaluate is the second, so we next do that:

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

Note that we no longer need to consider the third row on the table: it will not be a bad row, because (at least) one of premises is false on that row. 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 rows, so this claimed entailment is genuine. (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, this time for validity. Let us check whether the following argument is valid

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

So we need to check whether the premises entail the conclusion.

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 row apart from the few rows where the conclusion is false.

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

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

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

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

By the notion of a *bad* rows – a potential *counterexample* to a purported entailment – you can save yourself a huge amount of work in testing for validity. There is still lots of work involved in symbolising any natural language argument into Sentential, but once

that task is undertaken it is a relatively automatic process to determine whether the symbolisation is an entailment.

Key Ideas in §12

- Some shortcuts are available in constructing truth tables. For example, if a conjunction has one false conjunct, we needn't check the truth value of the other in order to determine that the whole conjunction is false.
- When applying our test for entailment, we need only check those rows on which all the premises are true to see if the conclusion is false on those rows. So we needn't check any row where the conclusion is true, or where a premise is false.

Practice exercises

A. Using shortcuts, determine whether each sentence is a logical truth, a logical falsehood, or neither.

1. $\neg B \land B$
2. $\neg D \lor D$
3. $(A \land B) \lor (B \land A)$
4. $\neg\big(A \to (B \to A)\big)$
5. $A \leftrightarrow \big(A \to (B \land \neg B)\big)$
6. $\neg(A \land B) \leftrightarrow A$
7. $A \to (B \lor C)$
8. $(A \land \neg A) \to (B \lor C)$
9. $(B \land D) \leftrightarrow \big(A \leftrightarrow (A \lor C)\big)$

13

Partial Truth Tables

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

13.1 Direct Uses of Partial Truth Tables

Logical Truth In order to show that a sentence is a *logical truth* (tautology), we need to show that it is true on every valuation. That is to say, we need to know that it comes out true on every row of the truth table. So, it seems, we need a complete truth table.

To show that a sentence is *not* a logical truth, however, we only need one valuation, corresponding to a truth table row on which the sentence is false. Therefore, in order to show that some sentence is not a logical truth, it is enough to provide a single valuation – a single row of the truth table – which makes the sentence false.

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

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

We have only left space for one row, rather than 16, since we are only looking for one valuation on which the sentence is false. For just that reason, we have filled in 'F' for the entire sentence. A partial truth table is a device for 'reverse engineering' a valuation, given a truth value assigned to a complex sentence. We work backward from that truth value to what the valuation must or could be.

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

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

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

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

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

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

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

Logical Falsehood Showing that something is a *logical falsehood* (contradiction) requires us to consider every row of a complete truth table. We need to show that there is no valuation which makes the sentence true; that is, we need to show that the sentence is false on every row of the truth table.

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

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

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

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

Equivalence To show that two sentences are *logically equivalent*, we must show that the sentences have the same truth value on every valuation. So this requires us to consider each row of a complete truth table.

To show that two sentences are *not* logically equivalent, we only need to show that there is a valuation on which they have different truth values. So this requires only a

partial truth table: construct the table given the assumption that one sentence is true and the other false. So, for example, to show that '$(\neg A \wedge B)$' and '$(A \vee \neg B)$' are not logically equivalent, constructing this partial truth-table would suffice:

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

Disjunctions and biconditionals in partial truth tables However, unless we are lucky, we might need to construct a partial truth table with two rows. (Or attempt to construct two partial truth tables.) If we are trying to show $\mathcal{A}$ and $\mathcal{B}$ logically *inequivalent*, we will want to consider the possibility that it is $\mathcal{A}$ which is true and $\mathcal{B}$ which is false, as well as the other possibility that it is $\mathcal{B}$ which is true and $\mathcal{A}$ which is false.

Suppose we are considering the sentence '$\neg(P \vee Q)$', and we are testing whether it is not a logical truth. So we begin our partial truth table by assuming it false and seeing if we can construct a valuation.

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

We see that the falsity of the whole negated sentence requires the embedded disjunction to be true. But what we do we now? There is no unique way to proceed. What we do is add a new row, in effect 'branching' the possible ways of constructing a valuation which makes this embedded disjunction true. On the first we assume the first disjunct is true, and on the second that the right disjunct is true

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

But we can then see that no matter how we fill in the blank cells, in either row, we will be able to make the whole sentence come out false. For example:

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

So it is possible to construct a valuation – more than one – on which this sentence is false, so '$\neg(P \vee Q)$' is not a logical truth.

Consistency To show that some sentences are *jointly consistent*, we must show that there is a valuation which makes all of the sentence true. So this requires only a partial truth table with a single row.

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

Validity To show that an argument is *valid*, we must show that there is no valuation which makes all of the premises true and the conclusion false. So this requires us to consider all valuations in a complete truth table.

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

This table summarises what we need to consider in order to demonstrate the presence or absence of various semantic features of sentences and arguments. So, checking a sentence for contradictoriness involves considering all valuations, and we can directly do that by constructing a complete truth table.

	Check if yes	**Check if no**
logical truth?	all valuations: complete truth table	one valuation: partial truth table
logical falsehood?	all valuations: complete truth table	one valuation: partial truth table
logically equivalent?	all valuations: complete truth table	one valuation: partial truth table
consistent?	one valuation: partial truth table	all valuations: complete truth table
valid?	all valuations: complete truth table	one valuation: partial truth table
entailment?	all valuations: complete truth table	one valuation: partial truth table

In all these uses of partial truth tables, we must begin constructing them with a particular semantic property in mind. We will begin the construction with a different hypothesis about the target sentence, depending on what property we are testing for. If we are using partial truth tables to test consistency, we will begin by assigning each sentence 'true'. If we are testing for validity, we will assign the premises 'true', but the conclusion 'false'.

13.2 Indirect Uses of Partial Truth Tables

We just saw how to use partial truth tables to directly construct a valuation which demonstrates that an argument is invalid, or that some sentences are consistent, etc.

But it turns out we can use the method of partial truth tables in an indirect way to also evaluate arguments for validity or sentences for inconsistency. The idea is this: we attempt to construct a partial truth table showing that the argument is invalid, and *if we fail*, we can conclude that the argument is in fact valid.

So consider showing that an argument is invalid, which we just saw requires only a one-line partial truth table on which all of the premises are true and the conclusion is false. So consider an attempt to show this argument invalid: $(P \wedge R), (Q \leftrightarrow P) \therefore Q$. We construct a partial truth table, and attempt to construct a valuation which makes all the premises true and the conclusion false:

P	Q	R	$(P \wedge R)$	$(Q \leftrightarrow P)$	Q
	F		**T**	**T**	**F**

Looking at the second premise, if we are to construct this valuation we need to make P false: the premise is true, so both constituents have to have the same truth value, and Q is false by assumption in this valuation:

P	Q	R	$(P \wedge R)$	$(Q \leftrightarrow P)$	Q
	F		**T**	F **T** F	**F**

But looking at the first premise, we see that both P and R have to be true to make this conjunction true:

P	Q	R	$(P \wedge R)$	$(Q \leftrightarrow P)$	Q
??	F	T	T **T** T	F **T** F	**F**

The truth of the first premise (given the other assumptions) has to make P true, but the truth of the second (given the other assumptions) has to make P false. So: *there is no coherent way of assigning a truth value to P so as to make this argument invalid.* (This is marked by the '??' in the partial truth table.) Hence, it is valid.

I call this an INDIRECT use of partial truth tables. We do not construct the valuations which actually demonstrate the presence or absence of a semantic property of an argument or set of sentences. Rather, we show that the assumption that there is a valuation that meets a certain condition is not coherent. So in the above case, we conclude that nowhere among the 8 rows of the complete truth table for that argument is one that makes the premises true and the conclusion false.

This procedure works because our partial truth table test is guaranteed to succeed in demonstrating the absence of validity, for an invalid argument. Accordingly, if our test fails to demonstrate the absence of validity, that must be because the argument is in fact valid. (This is in keeping with the table at the end of the previous section. Our failure to construct a valuation showing the argument invalid implicitly considers all valuations.)

This indirect method of partial truth tables can also be used if we need to add additional branches to our partial truth table. Suppose we are testing if '$P \leftrightarrow \neg P$' is a logical falsehood. We attempt to construct a valuation making it true:

P	$(P \leftrightarrow \neg P)$
	T

Again we need to branch our partial truth table to deal with the two possible ways this biconditional might be true: if the two sides are both false, or if they are both true

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

We can see, as we complete the table, that there is no coherent valuation making this sentence true. So the indirect method allows us to deduce that it is a logical falsehood:

P	$(P \leftrightarrow \neg P)$
??	T **T** T F
??	F **T** F T

Key Ideas in §13

- A partial truth table is a way to reverse engineer a valuation, given a truth value for a sentence.
- Partial truth tables can be effective tests for the absence of most of the semantic properties of sentences and arguments; and they can provide a test for the presence of consistency in some sentences.
- We can also use partial truth tables indirectly to test for the presence of semantic properties of sentences and arguments, by showing that those properties cannot be absent.

Practice exercises

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

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

B. Use complete or partial truth tables (as appropriate) to determine whether these sentences are jointly consistent, or jointly inconsistent:

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

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

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

14

Expressiveness of Sentential

When we introduced the idea of truth-functionality in §8.2, we observed that every sentence connective in Sentential was truth-functional. As we noted, that property allows us to represent complex sentences involving only these connectives using truth tables.

14.1 Other Truth-Functional Connectives

Are there other truth functional connectives than those in Sentential? If there were, they would have schematic truth tables that differ from those for any of our connectives. And it is easy to see that there are. Consider this proposed connective:

The Sheffer stroke For any sentences $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \downarrow \mathcal{B}$ is true if and only if both $\mathcal{A}$ and $\mathcal{B}$ are false. We can summarize this in the schematic truth table for the Sheffer Stroke:

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

Inspection of the schematic truth tables for $\wedge$, $\vee$, etc., shows that their truth tables are different from this one, and hence the Sheffer Stroke is not one of the connectives of Sentential. It is a connective of English however: it is the 'neither … nor …' connective that features in 'Siya is neither an archer nor a jockey', which is false iff she is either.

'Whether or not' The connective '… whether or not …', as in the sentence 'Sam is happy whether or not she's rich' seems to have this schematic truth table:

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A}$ whether or not $\mathcal{B}$
T	T	T
T	F	T
F	T	F
F	F	F

This too corresponds to no existing connective of Sentential.

In fact, it can be shown that there are many other potential truth-functional connectives that are not included in the language Sentential.[1]

14.2 The Expressive Power of Sentential

Should we be worried about this, and attempt to add new connectives to Sentential? It turns out we already have enough connectives to say *anything* we wish to say that makes use of only truth-functional connectives.[2] The connectives that are already in Sentential are TRUTH-FUNCTIONALLY COMPLETE:

> For any truth-functional connective in any language – that is, one which has a truth-table like ours – there is a schematic sentence of Sentential which has the same truth table.

Remember that a schematic sentence is something like $\mathcal{A} \vee \neg\mathcal{B}$, where arbitrary Sentential sentences can fill the places indicated by $\mathcal{A}$ and $\mathcal{B}$.

This is actually not very difficult to show. We'll start with an example. Suppose we have the English sentence 'Siya is exactly one of an archer and a jockey'. This sentence features the connective 'Exactly one of ... and ...'. In this case, the simpler sentences which are connected to form the complex sentence are 'Siya is an archer' and 'Siya is a jockey'. The schematic truth table for this connective is as follows:

$\mathcal{A}$	$\mathcal{B}$	'Exactly one of $\mathcal{A}$ and $\mathcal{B}$'
T	T	**F**
T	F	**T**
F	T	**T**
F	F	**F**

[1] There are in fact sixteen truth-functional connectives that join two simpler sentences into a complex sentence, but Sentential only includes four. (Why sixteen? Because there are four rows of the schematic truth table for such a connective, and each row can have either a T or an F recorded against it, independent of the other rows, so there are $2 \times 2 \times 2 \times 2 = 16$ ways of constructing such a truth-table.)

[2] I should flag a potential limitation here: this result needs our assumption that True and False are the only truth values. Some MANY-VALUED LOGICS include further 'truth values'; for example a third truth value Indeterminate. (It is dubious whether that is a truth value, or whether a sentence having it just reflects our ignorance of its truth or falsity.) Such logics can have truth-functional connectives that don't behave like any connective of Sentential. One motivating example for a third truth value, neither true nor false, are cases of presupposition failure we will look at in §19.4.

We want now to find a schematic Sentential sentence that has this same truth table. So we shall want the sentence to be true on the second row, and true on the third row, and false on the other rows. In other words, we want a sentence which is true on *either* the second row *or* the third row.

Let's begin by focusing on that second row, or rather the family of valuations corresponding to it. Those valuations include only those that make $\mathcal{A}$ true and and $\mathcal{B}$ false. These are the only valuations among those we are considering which make $\mathcal{A}$ true and $\mathcal{B}$ false. So they are the only valuations which make both $\mathcal{A}$ and $\neg\mathcal{B}$ true. So we can construct a sentence which is true on valuations in that family, and those valuations alone: the conjunction of $\mathcal{A}$ and $\neg\mathcal{B}$, $(\mathcal{A} \wedge \neg\mathcal{B})$.

Now look at the third row and its associated family of valuations. Those valuations make $\mathcal{A}$ false and $\mathcal{B}$ true. They are the only valuations among those we are considering which make $\mathcal{A}$ false and $\mathcal{B}$ true. So they are the only valuations among those we are considering which make both of $\neg\mathcal{A}$ and $\mathcal{B}$ true. So we can construct a schematic sentence which is true on valuations in that family, and only those valuations: the conjunction of $\neg\mathcal{A}$ and $\mathcal{B}$, $(\neg\mathcal{A} \wedge \mathcal{B})$.

Our target sentence, the one with the same truth table as 'Exactly one of $\mathcal{A}$ and $\mathcal{B}$', is true on either the second or third valuations. So it is true if either $(\mathcal{A} \wedge \neg\mathcal{B})$ is true or if $(\neg\mathcal{A} \wedge \mathcal{B})$ is true. And there is of course a schematic Sentential sentence with just this profile: $(\mathcal{A} \wedge \neg\mathcal{B}) \vee (\neg\mathcal{A} \wedge \mathcal{B})$.

Let us summarise this construction by adding to our truth table:

$\mathcal{A}$	$\mathcal{B}$	'Exactly one of $\mathcal{A}$ and $\mathcal{B}$'	$(\mathcal{A} \wedge \neg\mathcal{B})$	$\vee$	$(\neg\mathcal{A} \wedge \mathcal{B})$
T	T	**F**	F	**F**	F
T	F	**T**	T	**T**	F
F	T	**T**	F	**T**	T
F	F	**F**	F	**F**	F

As we can see, we have come up with a schematic Sentential sentence with the intended truth table.

14.3 The Disjunctive Normal Form Procedure

The procedure sketched above can be generalised:

1. First, identify the truth table of the target connective;
2. Then, identify which families of valuations (schematic truth table rows) the target sentence is true on, and for each such row, construct a conjunctive schematic Sentential sentence true on that row alone. (It will be a conjunction of schematic letters sentences of those schematic letters which are true on the valuation, and negated schematic letters, for those schematic letters false on the valuation).
   - What if the target connective is true on no valuations? Then let the schematic Sentential sentence $(\mathcal{A} \wedge \neg\mathcal{A})$ represent it – it too is true on no valuation.

3. Finally, the schematic Sentential sentence will be a disjunction of those conjunctions, because the target sentence is true according to any of those valuations.

   › What if there is only one such conjunction, because the target sentence is true in only one valuation? Then just take that conjunction to be the Sentential rendering of the target sentence.

Logicians say that the schematic sentences that this procedure spits out are in DISJUNCTIVE NORMAL FORM.

This procedure doesn't always give the simplest schematic Sentential sentence with a given truth table, but for any truth table you like this procedure gives us a schematic Sentential sentence with that truth table. Indeed, we can see that the Sentential sentence $\neg(\mathcal{A} \leftrightarrow \mathcal{B})$ has the same truth table as our target sentence too.

The procedure can be used to show that there is some redundancy in Sentential itself. Take the connective $\leftrightarrow$. Our procedure, applied to the schematic truth table for $\mathcal{A} \leftrightarrow \mathcal{B}$, yields the following schematic sentence:

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

This schematic sentence says the same thing as the original schematic sentence with the biconditional as its main connective, without using the biconditional. This could be used as the basis of a program to remove the biconditional from the language. But that would make Sentential more difficult to use, and we will not pursue this idea further.

Key Ideas in §14

› There are truth-functional connectives, such as 'neither ... nor ...', which don't correspond to any of the official connectives of Sentential.

› Nevertheless for any sentence structure $\mathcal{A} \oplus \mathcal{B}$, where '$\oplus$' is a truth-functional connective, it is possible to construct a Sentential schematic sentence which has the same truth table. (Indeed, one can do this making use only of negation, conjunction, and disjunction.)

› So Sentential is in fact able to express any truth-functional connective.

Practice exercises

A. For each of columns (i), (ii) and (iii) below, use the procedure just outlined to find a Sentential sentence that has the truth table depicted:

$\mathcal{A}$	$\mathcal{B}$	(i)	(ii)	(iii)
T	T	T	T	F
T	F	T	T	T
F	T	T	F	F
F	F	T	T	T

B. Can you find Sentential schematic sentences which have the same truth table as these English connectives?

1. '$\mathcal{A}$ whether or not $\mathcal{B}$';
2. 'Not both $\mathcal{A}$ and $\mathcal{B}$';
3. 'Neither $\mathcal{A}$ nor $\mathcal{B}$, but at least one of them';
4. 'If $\mathcal{A}$ then $\mathcal{B}$, else $\mathcal{C}$'.

Chapter 4

The Language of Quantified Logic

15

Building Blocks of Quantifier

15.1 The Need to Decompose Sentences

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

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

To symbolise it in Sentential, we might offer a symbolisation key:

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

And the argument itself becomes:

$$L, A \therefore F$$

This is *invalid* in Sentential – there is a valuation on which the premises are true and the conclusion false. But the original English argument is clearly valid.

The problem is not that we have made a mistake while symbolising the argument. This is the best symbolisation we can give in Sentential. The problem lies with Sentential itself. 'All logicians wear funny hats' is about both logicians and hat-wearing. By not retaining this in our symbolisation, we lose the connection between Willard's being a logician and Willard's wearing a hat.

Another example. This argument is also intuitively valid:

John loves James;
So: John loves someone.

Again, the best Sentential symbolisation is the invalid $P \therefore Q$. The validity of this argument depends on the internal structure of the sentences, and specifically the connection between the name 'James' and the phrase 'someone'.

The basic units of Sentential are atomic sentences, and Sentential cannot decompose these. None of the sentences in the arguments above have any truth-functional connectives, so must be symbolised as atomic sentences of Sentential. To symbolise arguments like the preceding, we will have to develop a new logical language which will allow us to *split the atom*. We will call this language Quantifier, and the study of this language and its features is *quantified logic*.

The details of Quantifier will be explained throughout this chapter, but here is the basic idea about how to split the atom(ic sentence). The key insight is that many natural language sentences have *subject-predicate* structure, and some arguments are valid in virtue of this structure. Quantifier adds to Sentential some resources for modelling this structure – or perhaps more accurately, it allows us to model the predicate-name structure of many sentences, along with any truth-functional structure.

Names First, we have *names*. In Quantifier, we indicate these with lower case italic letters. For instance, we might let 'b' stand for Bertie, or let 'i' stand for Willard. The names of Quantifier correspond to proper names in English, like 'Willard' or 'Elyse', which also stand for the things they name.

Predicates Second, we have *predicates*. English predicates are expressions like '_____ is a dog' or '_____ is a logician'. These are not complete sentences by themselves. In order to make a complete sentence, we need to fill in the gap. We need to say something like 'Bertie is a dog' or 'Willard is a logician'. In Quantifier, we indicate predicates with upper case italic letters. For instance, we might let the Quantifier predicate 'D' symbolise the English predicate '_____ is a dog'. Then the expression 'Db' will be a sentence in Quantifier, which symbolises the English sentence 'Bertie is a dog'. Equally, we might let the Quantifier predicate 'L' symbolise the English predicate '_____ is a logician'. Then the expression 'Li' will symbolise the English sentence 'Willard is a logician'.

Quantifiers Third, we have quantifier phrases. These tell us *how much*. In English there are lots of quantifier phrases. But in Quantifier we will focus on just two: 'all'/'every' and 'there is at least one'/'some'. So we might symbolise the English sentence 'there is a dog' with the Quantifier sentence '$\exists x Dx$', which we would naturally read aloud as 'there is at least one thing, x, such that x is a dog'.

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

15.2 Singular Terms

In English, a SINGULAR TERM is a noun phrase that refers to a *specific* person, place, or thing. The word 'dog' is not a singular term, because there are a great many dogs. The phrase 'Bertie' is a singular term, because it refers to a specific terrier. Here are some further examples:

- **Proper Names**, e.g., 'Bertie enjoys playing fetch';
  'Scott Morrison breached the human rights of children seeking asylum';
- **Definite Descriptions**, e.g., 'The oldest person is a woman';
  'Ortcutt is the shortest spy';
- **Possessives**, e.g., 'Antony's eldest child loves coding';
- **Pronouns**, e.g., 'She *(points)* plays violin';
  'Icecream is delicious. Everyone loves it';
- **Demonstratives**, e.g., 'That loud dog is so annoying'.

The clearest cases of singular terms are proper names, and these occupy a distinct syntactic category in Quantifier. Most of the other types of English singular terms are modelled in more or less indirect ways in Quantifier.

Definites and possessives are handled principally by paraphrase. We treat possessives as disguised definite descriptions, paraphrasing 'Antony's eldest child' as something like 'the eldest child of Antony'. In turn, definites are handled as complex constructions in Quantifier, as we'll see when we take them up in §19.

Even trickier in some ways are singular term uses of pronouns and demonstratives. Both of these constructions rely on the CONVERSATIONAL CONTEXT to fix a determinate reference. I might need to gesture, or rely on our previous utterances, to understand who 'she' refers to in 'she plays violin'. Likewise, 'that loud dog' might vary in which dog it refers to from conversation to conversation. There are approaches that attempt to model the role of context in determining the meaning of an expression, but we will not attempt to do this here. Quantifier is not designed to model every aspect of English. If we are forced to try to represent some English sentences involving context-sensitive singular terms in Quantifier, we will need to resort to paraphrases that are not fully adequate (e.g., treating demonstratives as definite descriptions, despite their differences).

Moreover, pronouns are not singular terms in every use. Compare the uses of the pronoun 'she' in 'She plays violin' and 'Every girl thinks she deserves icecream'. The first refers to a specific individual, perhaps with some contextual cues like gestures to help identify which. The second, however, doesn't refer to a specific girl – rather, it ranges over all girls. Perhaps surprisingly, Quantifier takes this second kind of use of pronouns to be primary. We will discuss how Quantifier represents pronouns when we introduce variables in §15.5.

15.3 Names

PROPER NAMES are a particularly important kind of singular term. These are expressions that label individuals without describing them. The name 'Emerson' is a proper name, and the name alone does not tell you anything about Emerson. Of course, some names are traditionally given to boys and other are traditionally given to girls. If 'Hilary' is used as a singular term, you might guess that it refers to a woman. You might,

though, be guessing wrongly. Indeed, the name does not necessarily mean that what is referred to is even a person: Hilary might be a giraffe, for all you could tell just from the name 'Hilary'. In English, the use of certain names triggers our knowledge of these conventions, so that (for example) the use of name 'Fido' might well trigger an expectation that the thing named is a dog. However, while it would violate convention to name your child 'Fido', once the you manage to assign the name it can perfectly well refer to a human child.

In Quantifier, there are no conventions around its category of NAMES. These are pure names, whose only role is to designate some specific individual. Names in Quantifier are represented by lower-case letters 'a' through to 'r'. We can add subscripts if we want to use some letter more than once, if we have a complicated discourse with many different names. So here are some names in Quantifier:

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

These should be thought of along the lines of proper names in English. But with some differences. First, 'Antony Eagle' is a proper name, but there are a number of people with this name. (Equally, there are at least two people with the name 'P.D. Magnus' and several people named 'Tim Button'.) We live with this kind of ambiguity in English, allowing context to determine that some particular utterance of the name 'Antony Eagle' refers to one of the contributors to this book, and not some other guy. In Quantifier, 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.) Second, names (and predicates) in Quantifier are assigned their meaning, or *interpreted*, only temporarily. (This is just like the way that atomic sentences are only assigned a truth value in Sentential temporarily, relative to a valuation.)

As with Sentential, we provide symbolisation keys. These indicate, temporarily, what a name shall pick out. So we might offer:

e: Elsa
g: Gregor
m: Marybeth

Again, what we are saying in using natural language names is that the thing referred to by the Quantifier name 'e' is stipulated – for now – to be the same thing referred to by the English proper noun 'Elsa'. We are *not* saying that 'e' and 'Elsa' are synonyms. For all we know, perhaps there is additional nuance to the meaning of the name 'Elsa' other than what it refers to. If there is, it is not preserved in the Quantifier name 'e', which has no nuance in its meaning other than the thing it denotes. The Quantifier name 'e' might be stipulated to denote Elsa on one occasion, and Eddie on another.

You may have been taught in school that a noun is a 'naming word'. It is safe to use names in Quantifier to symbolise proper nouns. But dealing with common nouns is a more subtle matter. Even though they are often described as 'general names', common nouns actually function as names relatively rarely. Some common nouns which do often function as names are NATURAL KIND TERMS like 'gold' and 'tiger'. These are common nouns which really do name a kind of thing. Consider this example:

Gold is scarce;
Nothing scarce is cheap;
So: Gold isn't cheap.

This argument is valid in virtue of its form. The form of the first premise has the phrase '_____ is scarce' being predicated of 'gold', in which case, 'gold' must be functioning as a name in this argument. But notice that '_____ is gold' is a perfectly good predicate. So we cannot simply treat all natural kind terms as proper names of general kinds.

15.4 Predicates

The simplest predicates denote *properties* of individuals. They are things you can say about the features or behaviour of an object. Here are some examples of English predicates:

- _____ runs
- _____ is a dog
- _____ is a member of Monty Python
- _____ was a student of David Lewis
- A piano fell on _____

In general, you can think about predicates as things which combine with singular terms to make sentences. In these cases, they are interpreted with the aid of natural language VERB PHRASES. The most elementary phrases that correspond to predicates are simple intransitive verb phrases like 'runs'. But predicates can symbolise more complex verb phrases. In a subject-predicate sentence, we can treat any syntactic sub-unit of a sentence including everything but the subject as a predicate. So 'Bertie is a dog' can be seen as involving the name 'Bertie' and the predicate 'is a dog'. Note that proper names can occur within a predicate, as in the verb phrase 'was a student of David Lewis'. (We will see how to model sentences where multiple names interact with a complex predicate below in §17.) You can begin with sentences and make predicates out of them by removing singular terms and leaving 'slots' in which singular terms can be placed. Consider the sentence, 'Vinnie borrowed the family car from Nunzio'. By removing one singular term, we can obtain any of three different predicates:

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

› Vinnie borrowed the family car from _____

(What if we wanted to remove two or more singular terms and leave more than one gap? We shall return to this in §17.) Quantifier predicates are capital letters A through Z, with or without subscripts. We might write a symbolisation key for predicates thus:

A: _____ is angry

H: _____ is happy

If we combine our two symbolisation keys, we can start to symbolise some English sentences that use these names and predicates in combination. For example, consider the English sentences:

80. Elsa is angry.
81. Gregor and Marybeth are angry.
82. If Elsa is angry, then so are Gregor and Marybeth.

To make a simple subject-predicate sentence, we need to couple the predicate with as many names as there are 'gaps'. Since in the above key 'A' has one gap in the symbolisation key, following it with a single name makes a grammatically formed sentence of Quantifier. So sentence 80 is straightforward: we symbolise it by 'Ae'.

Sentence 81: this is a conjunction of two simpler sentences. The simple sentences can be symbolised just by 'Ag' and 'Am'. Then we help ourselves to our resources from Sentential, and symbolise the entire sentence by '$Ag \wedge Am$'. This illustrates an important point: Quantifier has all of the truth-functional connectives of Sentential.

Sentence 82: this is a conditional, whose antecedent is sentence 80 and whose consequent is sentence 81. So we can symbolise this with '$Ae \rightarrow (Ag \wedge Am)$'.

Predicates without subjects Actually we can make an elegant further assumption to incorporate Sentential entirely within Quantifier. A predicate combines with singular terms to make sentences. But some English predicates, though they require a singular term syntactically, don't seem to involve the referent of those singular terms in their meaning. We see this in expressions like these:

83. It is snowing;
84. It seems that George is hungry;
85. She'll be right.

The pronouns in these cases are known as DUMMY PRONOUNS. These pronouns have no obvious referent, but that poses no problem for the interpretation of the sentences. Example 83 means something like the bare verb 'snowfalling', if only that were grammatical. In these cases, the predicate following the pronoun does not denote a property of individuals, because there is no way to attach it to a particular individual. Contrast these examples:

86. Coober Pedy is a harsh place. It has underground houses.
87. Coober Pedy is a harsh place. It seldom rains there.

In 86, 'it' refers to Coober Pedy. Not so in 87: 'It' in that example is a DUMMY PRONOUN. Yet 'there' does refer to Coober Pedy. Drop the latter pronoun, and you obtain 'it seldom rains'. Having already argued that 'it' is a dummy pronoun in the longer sentence, it appears to remain a dummy pronoun in the shortened sentence. Try substituting some proper name for 'it' in 'it seldom rains' and see what nonsense you get.

Such predicates, where the dummy pronoun should not be thought of as a genuine gap in the sentence, and with no other singular terms evident, have a meaning which is purely determined by the predicate. English requires a grammatical subject, but Quantifier is not subject to a similar limitation. Rather than reproduce this feature of English, we will allow Quantifier predicates occuring by themselves to count as grammatical sentences of Quantifier. These zero-place predicates, semantically requiring no subject, are to be symbolised by capital letters 'A' through 'Z' (perhaps subscripted), but without needing any adjacent name to be grammatical.

Note we have thereby included all atomic sentences of Sentential in Quantifier among these special predicates of Quantifier. In Sentential we used these sentences to symbolise any sentence of English which did not include sentence connectives. In Quantifier their use will be more limited. But nevertheless, syntactically, since Quantifier has all the connectives of Sentential and all the atomic sentences, we see that *every sentence of* Sentential *is also a sentence of* Quantifier.

15.5 Quantifiers

We are now ready to introduce quantifiers. In general, a quantifier tells us *how many*. Consider these sentences:

88. Everyone is happy.
89. Someone is angry.
90. Every girl thinks she deserves icecream.
91. Most people are happy.
92. Exactly two people are angry.
93. More than three people are happy.

We will focus initially on the coarse-grained quantifiers 'every'/'all' and 'some'. We will look at numerical quantifiers, as in examples 92 and 93, in §18.

It might be tempting to symbolise sentence 88 as '$(He \wedge (Hg \wedge Hm))$'. Yet this would only say that Elsa, Gregor, and Marybeth are happy. We want to say that *everyone* is happy, even those we have not named, even those who are nameless.

Note that 88 and 89 and 90 can be roughly paraphrased like this:

94. Every person is such that: they are happy.
95. Some person is such that: they are angry.
96. Every girl is such that: she thinks she herself deserves icecream.

In each of these, we have a pronoun – singular 'they' in 94 and 95, 'she' in 96 – which is governed by the preceding phrase. That phrase gives us information about what this pronoun is pointing to – is it pointing severally to everyone, as in 94? Or to just someone, though it is generally unknown which particular person it is, as in 95? In either case, something general is being said, rather than something specific, even in example 95 which is true just in case there is at least one angry person – it doesn't matter which person it is.

In this sort of construction, the sentences 'they are happy' and 'she thinks she herself deserves icecream', which are headed by a bare pronoun, are called OPEN SENTENCES. An open sentence in English can be used to say something meaningful, if the circumstances permit a unique interpretation of the pronoun – consider 'she plays violin' from §15.3. But in many cases no such unique interpretation is possible. If I gesture at a large crowd and say simply 'he is angry', I may not manage to say anything meaningful if there is no way to establish which person this use of 'he' is pointing to. The other part of the sentence, the 'every person is such that ...' part, is called a QUANTIFIER PHRASE. The quantifier phrase gives us guidance about how to interpret the otherwise bare pronoun.

The treatment of quantifier phrases in Quantifier actually follows the structure of these paraphrases rather well. The Quantifier analogue of these embedded pronouns is the category of VARIABLE. In Quantifier, variables are italic lower case letters 's' through 'z', with or without subscripts. They combine with predicates to form open sentences of the form '$\mathcal{A}x$'. Grammatically variables are thus like singular terms. However, as their name suggests, variables do not denote any fixed individual. They will not be assigned a meaning by a symbolisation key, even temporarily. Rather, their role is to be governed by an accompanying quantifier phrase to say something general about a situation. In Quantifier, an open sentence combines with a quantifier to form a sentence. (Notice that I have here returned to the practice of using '$\mathcal{A}$' as a metavariable, from §7.)

Universal Quantifier The first quantifier from Quantifier we meet is the UNIVERSAL QUANTIFIER, symbolised '$\forall$', and which corresponds to 'every'. Unlike English, we always follow a quantifier in Quantifier by the variable it governs, to avoid the possibility of confusion. Putting this all together, we might symbolise sentence 88 as '$\forall xHx$'. The variable 'x' is serving as a kind of placeholder, playing the role that is allotted to the pronoun in the English paraphrase 94. The expression '$\forall x$' intuitively means that you can pick anyone to be temporarily denoted by 'x'. The subsequent 'Hx' indicates, of that thing you picked out, that *it* is happy. (Note that pronoun again.)

I should say that there is no special reason to use 'x' rather than some other variable. The sentences '$\forall xHx$', '$\forall yHy$', '$\forall zHz$', and '$\forall x_5Hx_5$' use different variables, but they will all be logically equivalent.

Existential quantifier To symbolise sentence 89, we introduce a second quantifier: the EXISTENTIAL QUANTIFIER, '$\exists$'. Like the universal quantifier, the existential quantifier requires a variable. Sentence 89 can be symbolised by '$\exists xAx$'. Whereas '$\forall xAx$' is read naturally as 'for all x, it (x) is angry', '$\exists xAx$' is read naturally as 'there is something, x, such that it (x) is angry'. Once again, the variable is a kind of placeholder; we could just as easily have symbolised sentence 89 with '$\exists zAz$', '$\exists w_{256}Aw_{256}$', or whatever.

Some more examples will help. Consider these further sentences:

97. No one is angry.
98. There is someone who is not happy.
99. Not everyone is happy.

Sentence 97 can be paraphrased as, 'It is not the case that someone is angry'. We can then symbolise it using negation and an existential quantifier: '$\neg\exists xAx$'. Yet sentence 97 could also be paraphrased as, 'Everyone is not angry'. With this in mind, it can be symbolised using negation and a universal quantifier: '$\forall x\neg Ax$'. Both of these are acceptable symbolisations. Indeed, it will transpire that, in general, $\forall x\neg\mathcal{A}$ is logically equivalent to $\neg\exists x\mathcal{A}$. 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 98 is most naturally paraphrased as, 'There is some x, such that x is not happy'. This then becomes '$\exists x\neg Hx$'. Of course, we could equally have written '$\neg\forall xHx$', which we would naturally read as 'it is not the case that everyone is happy'. And that would be a perfectly adequate symbolisation of sentence 99.

Quantifiers get their name because they tell us how many things have a certain feature. Quantifier allows only very crude distinctions: we have seen that we can symbolise 'no one', 'someone', and 'everyone'. English has many other quantifier phrases: 'most', 'a few', 'more than half', 'at least three', etc. Some can be handled in a roundabout way in Quantifier, as we will see: the numerical quantifier 'at least three', for example, we will meet again in §18. But others, like 'most', are simply unable to be reliably symbolised in Quantifier.

15.6 Domains

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

In order to eliminate this ambiguity, we will need to specify a DOMAIN. The domain is just the things that we are talking about. So if we want to talk about people in Chicago, we define the domain to be people in Chicago. We write this at the beginning of the symbolisation key, like this:

domain: people in Chicago

The quantifiers *range over* the domain. Given this domain, '$\forall x$' is to be read roughly as 'Every person in Chicago is such that ...' and '$\exists x$' is to be read roughly as 'Some person in Chicago is such that ...'.

In Quantifier, the domain must always include at least one thing. Moreover, in English we can conclude 'something is angry' when given 'Gregor is angry'. In Quantifier, then,

we shall want to be able to infer '$\exists xAx$' from 'Ag'. So we shall insist that each name must pick out exactly one thing in the domain. If we want to name people in places beside Chicago, then we need to include those people in the domain.

In permitting multiple domains, Quantifier follows the lead of natural languages like English. Consider an argument like this:

100. All the beer has been drunk; so we're going to the bottle-o.

The premise says that all the beer is gone. But the conclusion only makes sense if there is more beer at the bottle shop. So whatever domain of things we are talking about when we state the premise, it cannot include absolutely *everything*. In Quantifier, we sidestep the interesting issues involved in deciding just what domain is involved in evaluating sentences like 'all the beer has been drunk', and explicitly include the current domain of quantification in our symbolisation key.

Note further that to make sense of the sentence 'all the beer has been drunk', the domain will have to contain both past and present things, so we can understand what we are saying about the now-absent beer. A domain contains *what we are talking about*. It might be difficult to understand how we do it, but we do talk about past things, fictional things, abstract things, merely possible things, and other unusual entities. So our domains must be flexible enough to include any of these things we might be talking about. It is a question in philosophical logic as to how we can explain how we manage to include nonexistent things in our domain of discourse, but for the purposes of Quantifier all we need to know is that this is something we somehow manage to do.

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. The domain can consist of anything we might be discussing; it is not restricted to things that presently exist.

Key Ideas in §15

- Quantifier gives us resources for modelling some aspects of sub-sentential structure in natural languages: names, predicates, and some quantifier phrases.
- The names of Quantifier are expressions that are used to refer to objects in some circumscribed domain; they can often have their referents temporarily set by associating them with natural language proper names in a symbolisation key.
- The predicates of Quantifier are expressions that denote properties of objects; they can often have their referents temporarily set by associating them with natural language verb phrases in a symbolisation key.
- The quantifiers of Quantifier have a fixed meaning not set by a symbolisation key. They tell us how many things in the domain meet a certain predicate. In Quantifier, we concentrate on two quantifiers: the universal '∀' ('every') and existential '∃' ('some').
- Quantifiers are supported by variables, which behave rather like pronouns in English.

Practice exercises

A. In each of the following sentences, identify the names and predicates. Comment on any difficulties.

1. Kurt and Alonzo are logicians;
2. Silver is useful in film photography;
3. Mary's ring is silver;
4. Kurt and Alonzo lift a couch;
5. The biggest threat to life on earth is carbon dioxide.

B. Identify the possible predicates that can be found by replacing singular terms with gaps in these sentences:

1. He dislikes Joel;
2. Andrew Leigh was a professor of economics at the ANU;
3. The professor of genomics was elected president of the Academy.

C. Make use of this symbolisation key to symbolise the following sentences into Quantifer, commenting on any difficulties:

domain: cities and towns in South Australia
a: Adelaide
m: Mount Gambier
T: _____ is a town
U: _____ is ugly
L: _____ is large

1. Adelaide is big and ugly.
2. Mount Gambier is not large.
3. If Mount Gambier is large, then Adelaide definitely is!
4. Mount Gambier is a large village.
5. Every city and town is ugly.

D. Can this argument be adequately symbolised in Quantifier? Comment on any difficulties.

1. She is tall;
2. Bob likes her;

So: Bob likes someone tall.

16

Sentences with One Quantifier

We now have the basic pieces of Quantifier. Symbolising many sentences of English will only be a matter of knowing the right way to combine predicates, names, quantifiers, and the truth-functional connectives. There is a knack to this, and there is no substitute for practice.

16.1 Common Quantifier Phrases

As in Sentential (recall §5), we will give canonical symbolisations for certain common English quantificational structures. Consider these sentences:

101. Every coin in my pocket is a 20¢ piece.
102. Some coin on the table is a dollar.
103. Not all the coins on the table are dollars.
104. None of the coins in my pocket are dollars.

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
Q: _____ is a 20¢ piece
D: _____ is a dollar

Sentence 101 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. So if we want to talk just about coins in my pocket, we will need to *restrict* the quantifier, by imposing a condition on the things we are saying are 20¢ pieces. That is: something in the domain is claimed to be a 20¢ piece *only if* it meets the restricting condition. That leads us to this conditional paraphrase:

105. For any (coin): $\underbrace{\textit{if}\text{ that coin is in my pocket}}_{\textit{restriction}}$, *then* it is a 20¢ piece.

So we can symbolise it as '$\forall x(Px \to Qx)$'.

Since sentence 101 is about coins that are both in my pocket *and* that are 20¢ pieces, it might be tempting to translate it using a conjunction. However, the sentence '$\forall x(Px \land Qx)$' would symbolise the sentence 'every coin is both a 20¢ piece and in my pocket'. This obviously means something very different than sentence 101. And so we see:

> A sentence can be symbolised as $\forall x(\mathcal{F}x \to \mathcal{G}x)$ if it can be paraphrased in English as 'every F is G' or 'all Fs are Gs'.

Example 102 uses the quantifier phrase 'some'. The same thought could be expressed using different quantifier phrases:

106. <u>At least one</u> coin on the table is a dollar.
107. <u>There is</u> a coin on the table that is a dollar.

These phrases all indicate an existential quantifier. In these examples, the class of coins on the table is being related to the class of dollar coins, and it is claimed that at least one member of the former class is also in the latter class – that there is *overlap*. This is represented in Quantifier following the example of this paraphrase:

108. There is something (a coin): it is in *both* the class of things on the table, *and* in the class of dollar coins.

That is symbolised using a conjunction, '$\exists x(Tx \land Dx)$'.

We know from Sentential that the order of conjuncts doesn't matter: '$(P \land Q)$' is logically equivalent to '$(Q \land P)$'. Likewise in Quantifier, '$\exists x(Tx \land Dx)$' is logically equivalent to '$\exists x(Dx \land Tx)$'. This fits well with the English sentences we are symbolising, because overlap is itself a symmetrical relation between classes. We see this also in the fact that we can paraphrase example 102 as 'Some dollar coin is on the table'.

Notice that we needed to use a conditional with the universal quantifier, but we used a conjunction with the existential quantifier. Suppose we had instead written '$\exists x(Tx \to Dx)$'. That would mean that there is some object in the domain such that *if* it is T, then it is also D. For this to be true, we just need something to not be T. So it is *very easy* for '$\exists x(Tx \to Dx)$' to be true. Given our symbolisation, it will be true if some coin is not on the table. Of course there is a coin that is not on the table: there are coins lots of other places. That is rather less demanding than the claim that something is both T and D.

A conditional will usually be the natural connective to use with a universal quantifier, but a conditional within the scope of an existential quantifier tends to say something very weak indeed. As a general rule of thumb, do not put conditionals in the scope of existential quantifiers unless you are sure that you need one.

> A sentence can be symbolised as $\exists x(\mathcal{F}x \wedge \mathcal{G}x)$ if it can be paraphrased in English as 'some F is G', 'at least one F is G', or 'There is an F which is G'.

Sentence 103 can be paraphrased as, 'It is not the case that every coin on the table is a dollar'. So we can symbolise it by '$\neg\forall x(Tx \rightarrow Dx)$'. You might look at sentence 103 and paraphrase it instead as, 'Some coin on the table is not a dollar'. You would then symbolise it by '$\exists x(Tx \wedge \neg Dx)$'. Although it is probably not immediately obvious yet, these two sentences are logically equivalent. (This is due to the logical equivalence between $\neg\forall x\mathcal{A}$ and $\exists x\neg\mathcal{A}$, mentioned in §15, along with the logical equivalence between $\neg(\mathcal{A} \rightarrow \mathcal{B})$ and $\mathcal{A} \wedge \neg\mathcal{B}$.)

Sentence 104 can be paraphrased as, 'It is not the case that there is some dollar in my pocket'. This can be symbolised by '$\neg\exists x(Px \wedge Dx)$'. It might also be paraphrased as, 'Everything in my pocket is a nondollar', and then could be symbolised by '$\forall x(Px \rightarrow \neg Dx)$'. Again the two symbolisations are logically equivalent. Both are correct symbolisations of sentence 104.

16.2 Empty Predicates

In §15, I 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 a domain is called an EMPTY predicate (relative to that domain). This is worth exploring.

Suppose we want to symbolise these two sentences:

109. Every monkey knows sign language
110. 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 109 can now be symbolised by '$\forall x(Mx \rightarrow Sx)$'. Sentence 110 can be symbolised as '$\exists x(Mx \wedge Sx)$'.

It is tempting to say that sentence 109 *entails* sentence 110. That is, we might think that it is impossible for it to be the case that every monkey knows sign language, without

it's also being the case that some monkey knows sign language. But this would be a mistake. It is possible for the sentence '$\forall x(Mx \rightarrow Sx)$' to be true even though the sentence '$\exists x(Mx \wedge Sx)$' is false.

How can this be? The answer comes from considering whether these sentences would be true or false *if there are no monkeys*. If there are no monkeys at all (in some domain), then '$\forall x(Mx \rightarrow Sx)$' would be *vacuously* true. Take the domain of reptiles. Look at the domain, and pick any monkey you like – it knows sign language![1] There is certainly no counterexample to the claim available in this domain. And because of the role of the conditional in our symbolisation, it turns out that a universally quantified claim with an unsatisfied restricting condition will also be true. In Quantifier, a universally quantified sentence of the form $\forall x\mathcal{A}x \rightarrow \mathcal{B}x$ is *false* only if we can find something which is $\mathcal{A}$ without being $\mathcal{B}$. If we can't find such a thing, perhaps because we can't find anything which is $\mathcal{A}$ in the first place, then the sentence will be true (since truth is just lack of falsity, and this sentence isn't false because we can't find a case that falsifies it). This derives ultimately from the feature of Sentential we have already acknowledged to be questionably analogous to English, namely, the fact that a conditional is false only if there is a counterexample, a case where the antecedent is true and the consequent false.

Another example will help to bring this home. Suppose we extend the above symbolisation key, by adding:

R: _____ is a refrigerator

Now consider the sentence '$\forall x(Rx \rightarrow Mx)$'. This symbolises 'every refrigerator is a monkey'. And this sentence is true, given our symbolisation key. This is counterintuitive, since we do not want to say that there are a whole bunch of refrigerator monkeys. It is important to remember, though, that '$\forall x(Rx \rightarrow Mx)$' is true iff any member of the domain that is a refrigerator is a monkey. Since the domain is *animals*, there are no refrigerators in the domain. Again, then, the sentence is *vacuously* true.

If you were actually dealing with the sentence 'All refrigerators are monkeys', then you would most likely want to include kitchen appliances in the domain. Then the predicate 'R' would not be empty and the sentence '$\forall x(Rx \rightarrow Mx)$' would be false. Remember, though, that a predicate is empty only *relative* to a particular domain.

> When $\mathcal{F}$ is an empty predicate relative to a given domain, a sentence $\forall x(\mathcal{F}x \rightarrow \ldots)$ will be vacuously true of that domain.

16.3 Picking a Domain

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

[1] Remember this is not a counterfactual claim (8.6); even if 'All monkeys know sign language' is vacuously true in the domain of reptiles, that wouldn't mean that 'Any monkey *would* know sign language' is true.

111. Every rose has a thorn.

We might offer this symbolisation key:

R: _____ is a rose

T: _____ has a thorn

It is tempting to say that sentence 111 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 trivially true. This is not what we want. To symbolise sentence 111 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 111 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 Quantifier. 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:

112. Every cowboy sings a sad, sad song.

Our domain must now include both all the roses (so that we can symbolise sentence 111) and all the cowboys (so that we can symbolise sentence 112). 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 111 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 112 with '$\forall x(Cx \rightarrow Sx)$'.

In general, 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'.

If you choose a narrow domain, you can make the task of symbolisation easier. If we are attempting to symbolise example 96, 'every girl thinks that she deserves icecream', we can pick the domain to be girls and then we only need to introduce a predicate '_____ thinks that they themselves deserve icecream', symbolised 'D'. The symbolisation is then the simple '$\forall x Dx$'. But our options are limited if the conversation goes on to talk about things other than girls. On the other hand, if you pick an expansive domain (such as everything whatsoever), you can always just impose an appropriate restriction. In this case, we could introduce the predicate 'G' to stand for '_____ is a girl', and symbolise the sentence as '$\forall x(Gx \rightarrow Dx)$'.

When choosing an expansive domain, you must take some care with implicitly restricted predicates. It would not be appropriate to paraphrase 'all the beer has been drunk' relative to a very expansive domain as 'For everything: if it is beer, it has been drunk'. To adequately capture the intent, we shall need to make the implicit contextual restriction of the predicate explicit, in something like this paraphrase 'For everything: if it is beer at the party, it has been drunk'.

16.4 The Utility of Paraphrase

When symbolising English sentences in Quantifier, it is important to understand the structure of the sentences you want to symbolise. What matters is the final symbolisation in Quantifier, and sometimes you will be able to move from an English language sentence directly to a sentence of Quantifier. 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 Quantifier.

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

domain: women
B: _____ is a bassist.
R: _____ is a rock star.
k: Kim Deal

Now consider these sentences:

113. If Kim Deal is a bassist, then she is a rock star.
114. If any woman is a bassist, then she is a rock star.

The same words appear as the consequent in sentences 113 and 114 ('... she is a rock star'), but they mean very different things (recall §15.5). To make this clear, it often helps to paraphrase the original sentences into a more unusual but clearer form.

Sentence 113 can be paraphrased as, 'Consider Kim Deal: if she is a bassist, then she is a rockstar'. The bare pronoun 'she' gets to denote Kim Deal because of our initial 'Consider Kim Deal' remark. This then says something about one particular person, and can obviously be symbolised as '$Bk \rightarrow Rk$'.

Sentence 114 gets a very similar paraphrase, with the same embedded conditional: 'Consider any woman: if she is a bassist, then she is a rockstar'. The difference in the 'Consider ...' phrase however forces a very different intepretation for the sentence as a whole. Replacing the English pronouns by variables, the Quantifier equivalent of a pronoun, we get this awkward quasi-English paraphrase: 'For any woman x, if x is a bassist, then x is a rockstar'. Now this can be symbolised as '$\forall x(Bx \to Rx)$'. This is the same sentence we would have used to symbolise 'Every woman who is a bassist is a rock star'. And on reflection, that is surely true iff sentence 114 is true, as we would hope.

Consider these further sentences, and let us consider the same interpretation as above, though in a domain of all people.

115. If anyone is a bassist, then Kim Deal is a rock star.
116. If anyone is a bassist, then they are a rock star.

The same words appear as the antecedent in sentences 115 and 116 ('If anyone is a bassist...'). But it can be tricky to work out how to symbolise these two uses. Again, paraphrase will come to our aid.

Sentence 115 can be paraphrased, 'If there is at least one bassist, then Kim Deal is a rock star'. It is now clear that this is a conditional whose antecedent is a quantified expression; so we can symbolise the entire sentence with a conditional as the main connective: '$\exists xBx \to Rk$'.

Sentence 116 can be paraphrased, 'For all people x, if x is a bassist, then x is a rock star'. Or, in more natural English, it can be paraphrased by 'All bassists are rock stars'. It is best symbolised as '$\forall x(Bx \to Rx)$', just like sentence 114.

The word 'any' is particularly tricky, because it can sometimes mean 'every' and sometimes 'at least one'! Think about the two occurrences of 'any' in this sentence:

117. Any student will be happy if they have any money.
    For every student: if there exists some money they possess, then they will be happy.

This can be symbolised '$\forall x(Sx \to (\exists y(My \land Pxy) \to Hx))$', where the first occurrence of 'any' is represented by a universal quantifier, and the second by an existential quantifier.

The moral is that the English words 'any' and 'anyone' should typically be symbolised using quantifiers. And if you are having a hard time determining whether to use an existential or a universal quantifier, try paraphrasing the sentence with an English sentence that uses words *besides* 'any' or 'anyone'.[2]

[2] The story about 'any' and 'anyone' is actually rather interesting. It is well-known to linguists that 'any' has at least two readings: so-called FREE CHOICE 'ANY', which is more or less like a universal quantifier ('Any friend of Jessica's is a friend of mine!'), and NEGATIVE POLARITY (NPI) 'ANY', which only occurs in 'negative' contexts, like negation ('I don't want any peas!'), where it functions more or less like an existential quantifier ('It is not the case that: there exist peas that I want').

Interestingly, the antecedent of a conditional is a negative environment (being equivalent to $\neg\mathcal{A} \lor \mathcal{C}$), and so we expect that 'any' in the antecedent of a conditional will have an existential interpretation.

16.5 Quantifiers and Scope

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

118. If everyone is a bassist, then Tim is a bassist
119. Everyone is such that, if they are a bassist, then Tim is a bassist.

To symbolise these sentences, I shall have to add a new name to the symbolisation key, namely:

b: Tim

Sentence 118 is a conditional, whose antecedent is 'everyone is a bassist'. So we will symbolise it with '$(\forall x Bx \rightarrow Bb)$'. This sentence is *necessarily* true: if *everyone* is indeed a bassist, then take any one you like – for example Tim – and he will be a bassist.

Sentence 119, by contrast, might best be paraphrased by 'every person x is such that, if x is a bassist, then Tim is a bassist'. This is symbolised by '$\forall x(Bx \rightarrow Bb)$'. And this sentence is false. Kim Deal is a bassist. So 'Bk' is true. But Tim is not a bassist, so 'Bb' is false. Accordingly, '$Bk \rightarrow Bb$' will be false. So '$\forall x(Bx \rightarrow Bb)$' will be false as well.

In short, '$(\forall x Bx \rightarrow Bb)$' and '$\forall x(Bx \rightarrow Bb)$' 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 Sentential(§6.3), and it will help to explain it in this way. We define quantifier scope officially in §20, but we also return to it in a preliminary way in §17.2.

In the sentence '$(\neg Bk \rightarrow Bb)$', the scope of '$\neg$' is just the antecedent of the conditional. We are saying something like: if 'Bk' is false, then 'Bb' is true. Similarly, in the sentence '$(\forall x Bx \rightarrow Bb)$', the scope of '$\forall x$' is just the antecedent of the conditional. We are saying something like: if 'B' is true of *everything*, then 'Bb' is also true.

In the sentence '$\neg(Bk \rightarrow Bb)$', the scope of '$\neg$' is the entire sentence. We are saying something like: '$(Bk \rightarrow Bb)$' is false. Similarly, in the sentence '$\forall x(Bx \rightarrow Bb)$', the scope of '$\forall x$' is the entire sentence. We are saying something like: '$(Bx \rightarrow Bb)$' is true of *everything*.

Scope can make a drastic difference in meaning. Reconsider these examples:

And it does: 'If anyone is home, they will answer the door' means something like: 'If someone is home, then that person will answer the door'. It does not mean 'If everyone is home, then they will answer the door'. This is what we see in 115.

But 116 is not a conditional – its main connective is a quantifier. So here free choice 'any' is the natural interpretation, so we use the universal quantifier.

We see the same thing with the quantifier 'someone'. In 'if someone is a bassist, Kim Deal is', someone gets symbolised by an existential quantifier in the scope of a conditional. But in 'If someone is a bassist, they are a musician' it should be symbolised by a universal taking scope over a conditional.

120. $(\overbrace{\forall x Bx}^{\text{Scope of '}\forall x\text{'}} \rightarrow Bb)$
If everything is B, then b is too. *Trivially true*

121. $\overbrace{\forall x(Bx \rightarrow Bb)}^{\text{Scope of '}\forall x\text{'}}$
All Bs are such that b is F. *False if b isn't F but there are some Fs*

The moral of the story is simple. When you are using quantifiers and conditionals, be very careful to make sure that you have sorted out the scope correctly.

16.6 Dealing with Complex Adjectives

When we encounter a sentence like

122. 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 122 as '$Wh \wedge Ch$'. But now consider:

123. Julia Gillard is a former prime minister.
124. Julia Gillard is prime minister.

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

F: _____ is former

P: _____ is Prime Minister

j: Julia Gillard.

Then we would symbolise 123 by '$Fj \wedge Pj$', and symbolise 124 by 'Pj'. That would however be a mistake, since that symbolisation suggests that the argument from 123 to 124 is valid, because the symbolisation of the premise does logically entail the symbolisation of the conclusion.

'White' is a INTERSECTIVE adjective, which is a fancy way of saying that the white Fs are among the Fs *and* among the white things: any white car is a car and white, just like any successful lawyer is a lawyer and successful, and a one tonne rhinoceros is both a rhino and a one tonne thing. But 'former' is a PRIVATIVE adjective, which means that any former F is not now among the Fs. Other privative adjectives occur in phrases such

as 'fake diamond', 'Deputy Lord Mayor', and 'mock trial'. When symbolising these sentences, you cannot treat them as a conjunction. So you will need to symbolise '_____ is a fake diamond' and '_____ is a diamond' using completely different predicates, to avoid a spurious entailment between them. The moral is: when you see an adjectivally modified predicate like 'white car', you need to ask yourself carefully whether the modifier is intersective, and can be symbolised as a conjunctive predicate, or not.[3]

Things are a bit more complicated, however. Recall this example from page 89:

> Daisy is a small cow.

We note that a small cow is definitely a cow, and so it seems we might treat 'small' as an intersective adjective. We might formalise this sentence like this: '$Sd \wedge Cd$', assuming this symbolisation key:

S: _____ is small

C: _____ is a cow

d: Daisy

But note that our symbolisation would suggest that this argument is valid:

125. Daisy is a small cow; so Daisy is small.

The symbolised argument, $Sd \wedge Cd \therefore Sd$, is clearly valid.

But the original argument 125 is not valid. Even a small cow is still rather large. (Likewise, even a short basketball player is still generally well above average height.) The point is that '_____ is a small cow' denotes the property something has when it is small *for a cow*, while '_____ is small' denotes the property of being a small *thing*. (In ordinary speech we tend to keep the 'for an F' part of these phrases silent, and let our conversational circumstances supply it automatically.) But neither should we treat 'small' as a nonintersective adjective. If we do, we will be unable to account for the valid argument 'Daisy is a small cow, so Daisy is a cow'.

The correct symbolisation key will thus be this, keeping the other symbols as they were:

S: _____ is small-for-a-cow

On this symbolisation key, the valid formal argument $Sd \wedge Cd \therefore Cd$ corresponds to this valid argument, as it should:

[3] This caution also applies to adjectives which are neither intersective nor privative, like 'alleged' in 'alleged murderer'. These ought not be symbolised by conjunction either.

Daisy is a cow that is small-for-a-cow; so Daisy is a cow.

Likewise, this rather unusual English argument turns out to be valid too:

Daisy is a cow that is small-for-a-cow; so Daisy is small-for-a-cow.

(Note that it can be rather difficult to hear the English sentence 'Daisy is small' as saying the same thing as the conclusion of this argument, 'Daisy is small for a cow', which explains why 'Daisy is a small cow, so Daisy is small' strikes us as invalid.)

If we take these observations to heart, there are many intersective adjectives which can change their meaning depending on what predicate they are paired with. Small-for-an-oil-tanker is a rather different size property than small-for-a-mouse, but in ordinary English we use the phrases 'small oil tanker' and 'small mouse' without bothering to make these different senses of 'small' explicit.

The way that 'small' behaves makes it a member of the class of SUBSECTIVE adjectives, as in 'poor dancer'. These are like intersective adjectives in that every poor dancer is a dancer (and every small cow is a cow). But the way that 'poor' behaves in this expression is such that we *cannot* conclude that a poor dancer is poor – they are bad at dancing, not necessarily financially disadvantaged. In these cases, the meaning of the modifying adjective is itself modified by the noun: in 'poor dancer', we get a distinctively dancing-related sense of 'poor'.

When symbolising, it is best to make these modified adjectives very explicit, generally introducing a new predicate to the symbolisation key to represent the. Doing so blocks the fallacious argument from 'Daisy is a small cow' to 'Daisy is small', where the natural sense of the conclusion is the generic size claim 'Daisy is small-for-a-thing'. (Likewise, symbolising '_____ is poor dancer' as '_____ is poor-for-a-dancer and _____ is a dancer'

bloocks the fallacious argument from 'Rupert Murdoch is a poor dancer' to 'Rupert Murdoch is poor'.)

The upshot is this: though you can symbolise 'Daisy is a small cow' as a conjunction, you will need to symbolise '_____ is a small cow' and '_____ is a small animal' using

different predicates of Quantifier to stand for the different appearances of 'small' – to symbolise 'small for a cow' and 'small for an animal'.

The overall message is not particularly specific: treat adjectives with care, and always think about whether a conjunction or some other symbolisation best captures what is going on in the English. There is no substitute for practice in developing a good sense of how symbolise arguments.

16.7 Generics

One final complication presents itself. In English, there seems to be a difference between these sentences:

126. Ducks lay eggs;
127. All ducks lay eggs.

The sentence in 127 is false: drakes and ducklings do not, for example. But nevertheless 126 seems to be true, for all that. That sentence lacks an explicit quantifier – it doesn't say '*all* ducks lay eggs'. It is what is known as a GENERIC claim: it shares a structure with examples like 'cows eat grass' or 'rocks are hard'. Generic claims concern what is typical or normal: the typical duck lays eggs, the typical rock is hard, the typical cow eats grass. Unlike universally quantified claims, generics are *exception-tolerant*: even if drakes don't lay eggs, still, ducks lay eggs.

We cannot represent this exception-tolerance very easily in Quantifier. The initial idea is to use the universal quantifier, but this will give the wrong results in some cases. For it will make this argument come out valid, when it should be ruled invalid:

> Ducks lay eggs. Donald Duck is a duck. So Donald Duck lays eggs.

One alternative idea that we can implement is that the word 'ducks' in 127 is referring to a natural kind, the species of ducks. So in fact rather than being a quantified sentence, it is in fact just a subject-predicate sentence, saying something more or less like 'The duck is an oviparous species'. This certainly works for some cases, such as 'Rabbits are abundant', where we have to be understood as saying something about the kind. (How could an individual be abundant?)

But this cannot handle every aspect of 'ducks lay eggs'. People do treat those generics as quantified, because they are often willing to conclude things about individuals given the generic claim. Given the information that ducks lay eggs, and that Wilhelmina is a duck, most people conclude that Wilhelmina lays eggs – thus apparently treating the generic as having the logical role of a universal quantifier.

The proper treatment of generics in English remains a wide-open question.[4] We will not delve into it further, but you should be careful when symbolising not to be drawn into the trap of unwarily treating every generic as a universal.

Key Ideas in §16

- Our quantifiers, together with truth-functional connectives, suffice to symbolise many natural language quantifier phrases including 'all', 'some', and 'none'.
- Judicious choice of domain can save you trouble when symbolising, but can also hamper your ability to symbolise all the sentences you'd like to.
- Paraphrasing natural language sentences can often reveal their quantifier structure, even if the surface form is misleading.
- Complexities in symbolising arise from adjectival modification of predicates, empty predicates (relative to a chosen domain), and generics – take care.

4 See, for example, Sarah-Jane Leslie and Adam Lerner (2016) 'Generic Generalizations', in Edward N Zalta, ed., *The Stanford Encyclopedia of Philosophy*, https://plato.stanford.edu/archives/win2016/entries/generics/.

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 Quantifier.

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 Quantifier:

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 Quantifier:

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 a 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 Quantifier. In each case, try to decide if the argument you have symbolized is valid.

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.

17

Multiple Generality

So far, we have only considered sentences that require simple predicates with just one 'gap', and at most one quantifier. Much of the fragment of Quantifier that focuses on such sentences was already discovered and codified into *syllogistic* logic by Aristotle more than 2000 years ago. The full power of Quantifier really comes out when we start to use predicates with many 'gaps' and multiple quantifiers. Despite first appearances, the discovery of how to handle such sentences was a very significant one. For this insight, we largely have the German mathematician and philosopher Gottlob Frege (1879) to thank.[1]

17.1 Many-place Predicates

All of the predicates that we have considered so far concern properties that can be attributed to objects might have by themselves, as it were. 'Herbie is white', 'Kim Deal is a bassist', etc., all talk about the features of a single individual. The associated predicates, such as '_____ is white', 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 concern the *relationship* between two things. Here are some examples of many-place predicates in English:

_____ loves _____

_____ is to the left of _____

[1] Frege (1879) *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle a. S.: Louis Nebert. Translated as 'Concept Script, a formal language of pure thought modelled upon that of arithmetic', by S. Bauer-Mengelberg in J. van Heijenoort, ed. (1967) *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, Cambridge, MA: Harvard University Press. The same logic was independently discovered by the American philosopher and mathematician Charles Sanders Peirce: see Peirce (1885) 'On the Algebra of Logic. A Contribution to the Philosophy of Notation' *American Journal of Mathematics* **7**, pp. 197–202. **Warning**: don't consult either of these original works, which will likely just confuse you. The present text is the result of more than a century of refinement in how to present the basic systems Frege and Peirce introduced, and is a lot more user friendly.

_____ is in debt to _____

_____ is supervised by _____

These are TWO-PLACE predicates. They need to be filled in with two terms (names or pronouns, most commonly) in order to make a sentence. Conversely, if we start with an English sentence containing many singular terms, we can remove two singular terms, to obtain different two-place predicates. Consider the sentence 'Vinnie borrowed the family car from Nunzio'. By deleting two singular terms, we can obtain any of three different two-place predicates:

Vinnie borrowed _____ from _____;

_____ borrowed the family car from _____;

_____ borrowed _____ from Nunzio.

And by removing all three singular terms, we obtain a THREE-PLACE predicate:

_____ borrowed _____ from _____.

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

Now there is a little problem with the above. I have used the same symbol, '_____', to indicate a gap formed by deleting a term from a sentence. However (as Frege emphasised), these are *different* gaps. To obtain a sentence, we can fill them in with the same term, but we can equally fill them in with different terms, and in various different orders. The following are all perfectly good sentences, obtained by filling in the gaps in '_____ loves _____', but they mean quite different things:

Karl loves Karl;
Karl loves Imre;
Imre loves Karl;
Imre loves Imre.

The point is that we need some way of keeping track of the gaps in predicates, so that we can keep track of how we are filling them in.

Another way to put the point: when it comes to two-(or more)-place predicates, sometimes the order matters. 'Shaq is taller than Jordan' doesn't mean the same thing as 'Jordan is taller than Shaq'. It matters whose name fills the first gap in the predicate, and whose name fills the second.

To keep track of the gaps, we shall label them. The labelling conventions I adopt are best explained by example. Suppose I want to symbolise the following sentences:

128. Karl loves Imre.
129. Imre loves himself.
130. Karl loves Imre, but not vice versa.
131. Karl is loved by Imre.

I will start with the following symbolisation key:

domain: people
i: Imre
k: Karl
L: $\underset{1}{\underline{\quad}}$ loves $\underset{2}{\underline{\quad}}$

- Sentence 128 will now be symbolised by 'Lki'.
- Sentence 129 can be paraphrased as 'Imre loves Imre'. It can now be symbolised by 'Lii'.
- Sentence 130 is a conjunction. We might paraphrase it as 'Karl loves Imre, and Imre does not love Karl'. It can now be symbolised by '$Lki \wedge \neg Lik$'.
- Sentence 131 might be paraphrased by 'Imre loves Karl'. It can then be symbolised by 'Lik'. Of course, this erases the difference in tone between the active and passive voice; such nuances are lost in Quantifier.

This last example highlights something important. Suppose we add to our symbolisation key the following:

M: $\underset{2}{\underline{\quad}}$ loves $\underset{1}{\underline{\quad}}$

Here, we have used the same English word ('loves') as we used in our symbolisation key for 'L'. However, we have swapped the order of the *gaps* around (just look closely at those little subscripts!) So 'Mki' and 'Lik' now *both* symbolise 'Imre loves Karl'. 'Mik' and 'Lki' now *both* symbolise 'Karl loves Imre'. Since love can be unrequited, these are very different claims. The moral is simple. When we are dealing with predicates with more than one place, we need to pay careful attention to the order of the places.

With these examples in hand, I can now give the official account of how we understand Quantifier symbolisations. Suppose we have an Quantifier expression $\mathcal{A}t_1, \ldots, t_k$, where each t_i is a name or a variable, symbolising a singular term, and where $\mathcal{A}$ symbolises a k-place predicate. The i-th term is to be interpreted as filling the gap labelled 'i'. So consider the following symbolisation key:

domain: places
a: Adelaide
b: Alice Springs
c: Coober Pedy
B: $\underset{2}{\underline{\quad}}$ is between $\underset{1}{\underline{\quad}}$ and $\underset{3}{\underline{\quad}}$
K: $\underset{1}{\underline{\quad}}$ is between $\underset{2}{\underline{\quad}}$ and $\underset{3}{\underline{\quad}}$

. Then if we want to symbolise 'Coober Pedy is between Adelaide and Alice Springs', we can do so using either '$Bacb$' or '$Kcab$'. The difference is in how the symbolisation key instructs us to fill the gaps we have established in the predicate as we take steps to represent it symbolically. There is no 'right' answer here: either can be good. The representation using 'B' graphically represents which item is between the other two in the syntax itself, while the representation using 'K' is more faithful to the original English.

Suppose we add to our symbolisation key the following:

S: $\underbrace{____}_{1}$ thinks only of $\underbrace{____}_{1}$
T: $\underbrace{____}_{1}$ thinks only of $\underbrace{____}_{2}$
a: Alice

As in the case of 'L' and 'M' above, the difference between these examples is only in how the gaps in the construction '... thinks only of ...' are labelled. In 'T', we have labelled the two gaps differently. They do not *need* to be filled with different names or variables, but there is always the potential to put different names in those different gaps. In the case of 'S', the gaps have the same label. In some sense, there is *only one* gap in this sentence, which is why the symbolisation key associates it with a one-place predicate – it means something like 'x thinks only of themself'. The second predicate is more flexible. Take something we can say with the predicate 'S', such as 'Sa', 'Alice thinks only of herself'. We can express pretty much the same thought using the two-place predicate 'T': 'Taa'.

We have introduced a potential ambiguity in our treatment of predicates. (See also §20.2.) There is nothing overt in our language that distinguishes the one-place predicate 'A' (such that 'Ab' is grammatical) from the two-place predicate 'A' (such that 'Ab' is ungrammatical, but 'Abk' is grammatical). We are, in effect, just letting context disambiguate how many argument places there are in a given predicate, by assuming that in any expression of Quantifier we write down, the number of names or variables following a predicate indicates how many places it has. We could introduce a system to disambiguate: perhaps adding a superscripted '1' to all one-place predicates, a superscripted '2' to all two-place predicates, etc. Then 'A^1b' is grammatical while 'A^1bk' is not; conversely, 'A^2b' is ungrammatical and 'A^2bk' is grammatical. This system of superscripts would be effective but cumbersome. We will thus keep to our existing practice, letting context disambiguate. What you should *not* do, however, is make use of the same capital letter to symbolise two different predicates in the same symbolisation key. If you do that, context will not disambiguate, and you will have failed to give an interpretation of the language at all.

17.2 Scope and Nested Quantifiers

Once we have two (or more) gaps in a predicate, we can fill them with different things. We've so far seen cases where multiple names are slotted into a many-place predicate. But we can also insert other terms, like variables. So to continue our example using

the predicate 'T', '$\underset{1}{\underline{\quad\quad}}$ thinks only of $\underset{2}{\underline{\quad\quad}}$', we can put a variable in the first gap, and a name in the second, if we wish: 'Txa'. This isn't a sentence, because no quantifer tells us how to understand that variable. (The sentence might be representing 'they think only of Alice', but without context there is no determinate referent for the pronoun 'they'.) Introduce a quantifier, and we have an interpretable sentence:

132. $\forall xTxa$;
     Everyone thinks only of Alice.

The fact that we can fill the two gaps of two-place predicates with different things, or even with the same thing, gives us an reason to favour the two-place predicate symbolisation of 'Alice thinks only of themself' as 'Taa'. That allows us to symbolise certain arguments that cannot be adequately symbolised using a one-place predicate. For example: 'Alice thinks only of herself; so there is someone who is the only person Alice thinks of'. The symbolisation of this argument might be: '$Taa \therefore \exists xTax$'. This might have some prospect of being valid, whereas '$Sa \therefore \exists xTax$' will not be valid.

The real power of many-place predicates comes when we consider examples in which *both* gaps in the predicate are filled by variables governed by different quantifiers. In cases where the quantifier expressions interact, we can express things we cannot say even when we allow logically complex combinations of one-quantifier sentences. With this power comes potential confusion too. So let's proceed carefully.

Consider the sentence 'everyone loves someone'. This illustrates our goal, as two quantifier expressions occur in this sentence: 'everyone' and 'someone'. But it also illustrates the potential pitfalls, as there is a possible ambiguity in this sentence. It might mean either of the following:

133. For every person, there is some person that they love
134. There is some particular person whom every person loves

It is fairly straightforward to see that these don't mean the same thing. The first would be true as long as everybody has somebody they love. One sort of case in which 133 is true is the cyclic central love triangle in *Twelfth Night*, where Viola loves Duke Orsino, the Duke loves Olivia, and Olivia loves Viola (who, disguised as a young man, is the Duke's go-between with Olivia).

In the *Twelfth Night* situation, 134 is not true. It could only be true if everybody loves the same person, e.g., if the Duke, Viola, *and Olivia herself* all love Olivia.

How can we symbolise these two different disambiguations of our original sentence? (Remember: one of the strengths of symbolic logic is that it is supposed to be able to clearly represent that which would be ambiguous in natural language.)

Let's paraphrase a little more formally as we step towards a fully symbolic representation. As our sentence has two quantifiers, I will use numbers to link pronouns in our paraphrase with the quantifier expressions which govern them. Using this device, our sentences can be paraphrased as follows:

135. Everyone$_1$ is such that there is someone$_2$ such that: they$_1$ love them$_2$.
136. There is someone$_2$ such that everyone$_1$ is such that: they$_1$ love them$_2$.

(Take a moment to convince yourself that these paraphrases suceed.)

You can see immediately that the difference in these paraphrases lies in the order of the quantifier expressions, and the remainder of the paraphrase, 'they$_1$ love them$_2$', is the same in each sentence. Using variables to symbolise pronouns, and choosing the variable 'x' for 'they$_1$' and 'y' for 'them$_2$', we can symbolise this 'Lxy', where 'L' stands for the two-place predicate '$\underset{1}{\underline{\quad\quad}}$ loves $\underset{2}{\underline{\quad\quad}}$'.

The quantifier order in the paraphrases governs how they interact. As we saw in §16.5, the scope of a quantifier is roughly the Quantifier expression in which that quantifier is the main connective. (Later on we will be a little more precise about the way that quantifier scope functions in Quantifier: see §20.) So in '$\forall x \exists y Lxy$', the scope of '$\forall x$' is the whole sentence, while the scope of '$\exists y$' is just '$\exists y Lxy$'. The following guides us in interpreting these 'nested' quantifiers, in which one falls in the scope of another:

> When one quantifier occurs in the scope of another, the narrower scope quantifier should be understood *with respect to* the value assigned to a variable by the wider scope quantifier.

Let's apply this to our example. In 135 'everyone' comes first, and 'someone' comes next. The intended interpretation is that this is true iff for any person x that you pick, with respect to that choice you can then find someone y who x loves. If you had chosen someone else as the value of x, then parasitic on that different choice you may end up needing to find a different value for y. Compare the reversed quantifier scope in 136. That is true iff there is someone y such that, with respect to that particular choice for y, any person x you pick, x loves y. With respect to a different initial choice for y, there may be values for x that do not satisfy x loves y, but as the initial choice is governed by an existential quantifier, that won't undermine the truth of the sentence. This gives us our two different symbolisations:

- Sentence 133 can be symbolised by '$\forall x \exists y Lxy$'. Return to our example love triangle between Duke Orsino, Viola, and Olivia. For any of the three people you might choose, you can find another person in the domain who they love. So sentence 133 is true.

- Sentence 134 is symbolised by '$\exists y \forall x Lxy$'. Sentence 134 is *not* true in the *Twelfth Night* situation. For each of the people in the domain, you can find someone who *doesn't* love them, and hence no one is universally beloved. If, instead, each person loved Olivia, then we could find someone (Olivia), such that everyone else we examined turns out to love them. In that case, 134 would be true.

This example, besides giving some indication of how to read sentences with multiple quantifiers, illustrates that quantifier scope matters a great deal. Indeed, the mistake

that arises when one illegitimately switches them around even has a special name: a *quantifier shift fallacy*. Here is a real life example from Aristotle:[2]

> Suppose, then, that [A] the things achievable by action have some end that we wish for because of itself, and because of which we wish for the other things, and that we do not choose everything because of something else – for if we do, it will go on without limit, so that desire will prove to be empty and futile[; c]learly, [B] this end will be the good, that is to say, the best good. (Aristotle, *Nichomachean Ethics* 1094^{a}18-22)

Setting aside Aristotle's subsidiary argument about desire, this argument seems to involve the following pattern of inference:

	Every action aims at some end which is desired because of itself.	(∀∃)
So:	There is end desired because of itself which is the aim of every action, the best good.	(∃∀)

This argument form is obviously invalid. It's just as bad as:[3]

	Every dog has its day.	(∀∃)
So:	There is a day for all the dogs.	(∃∀)

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

17.3 Stepping Stones to Symbolisation

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

domain: people and dogs
- D: $\underset{1}{\underline{\quad}}$ is a dog
- F: $\underset{1}{\underline{\quad}}$ is a friend of $\underset{2}{\underline{\quad}}$
- O: $\underset{1}{\underline{\quad}}$ owns $\underset{2}{\underline{\quad}}$
- g: Geraldo

And now let's try to symbolise these sentences:

[2] Note that it is hotly contested whether Aristotle actually commits a fallacy here, given the compressed nature of his prose. See, *inter alia*, J L Ackrill (1999), 'Aristotle on *eudaimonia*', pp. 57–77 in N Sherman, ed., *Aristotle's Ethics: Critical Essays*, Rowman & Littlefield.

[3] Thanks to Rob Trueman for the example.

137. Geraldo is a dog owner.
138. Someone is a dog owner.
139. All of Geraldo's friends are dog owners.
140. Every dog owner is the friend of a dog owner.
141. Every dog owner's friend owns a dog of a friend.

Sentence 137 can be paraphrased as, 'There is a dog that Geraldo owns'. This can be symbolised by '$\exists x(Dx \wedge Ogx)$'.

Sentence 138 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 137, except that it is not specifically about Geraldo. So we can symbolise sentence 138 by:

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

I 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 Quantifier sentence *nor* an English sentence: it uses bits of Quantifier ('$\exists$', '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 Quantifier sentence, a bit of rough-working-out.

Sentence 139 can be paraphrased as, 'Everyone who is a friend of Geraldo is a dog owner'. Using our stepping-stone tactic, we might write

$$\forall x\big(Fxg \rightarrow x \text{ is a dog owner}\big)$$

Now the fragment that we have left to deal with, 'x is a dog owner', is structurally just like sentence 137. But it would be a mistake for us simply to put 'x' in place of 'g' from our symbolisation of 137, yielding

$$\forall x\big(Fxg \rightarrow \exists x(Dx \wedge Oxx)\big).$$

Here we have a CLASH OF VARIABLES. The scope of the universal quantifier, '$\forall x$', is the entire conditional. But 'Dx' also falls within the scope of the existential quantifier '$\exists x$'. Which quantifier has priority and governs the interpretation of the variable? In Quantifier, if a variable x occurs in an Quantifier sentence, it is always governed by the quantifier which has the narrowest scope which includes that occurrence of x. So in the sentence above, the quantifier '$\exists x$' governs every occurrence of 'x' in '$(Dx \wedge Oxx)$'. Given this, the symbolisation does not mean what we intended. It says, roughly, 'everyone who is a friend of Geraldo is such that there is a self-owning dog'. This is not at all the meaning of the English sentence we are aiming to symbolise.

To provide an adequate symbolisation, then, we must avoid clashing variables. We can do this easily enough. There was no requirement to use 'x' as the variable in our symbolisation of 137, so we can easily choose some different variable for our existential quantifier. That will give us something like this, which adequately symbolises sentence 139:

$$\forall x\big(Fxg \rightarrow \exists z(Dz \wedge Oxz)\big).$$

Sentence 140 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

$$\forall x\big(x \text{ is a dog owner} \rightarrow \exists y(y \text{ is a dog owner} \land Fyx)\big)$$

Completing the symbolisation, we end up with

$$\forall x\big(\exists z(Dz \land Oxz) \rightarrow \exists y\big(\exists z(Dz \land Oyz) \land Fyx\big)\big)$$

Note that we have used the same variable, 'z', in both the antecedent and the consequent of the conditional, but that these are governed by two different quantifiers. This is ok: there is no potential confusion here, because it is obvious which quantifier governs each variable. We might graphically represent the scope of the quantifiers thus:

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

Even in this case, however, you might want to choose different variables for every quantifier just as a practical matter, preventing any possibility of confusion for your readers.

Sentence 141 is the trickiest yet. First we paraphrase it as 'For any x that is a friend of a dog owner, x owns a dog which is also owned by a friend of x'. Using our stepping-stone tactic, this becomes:

$$\forall x\big(x \text{ is a friend of a dog owner} \rightarrow x \text{ owns a dog which is owned by a friend of } x\big).$$

Breaking this down a bit more:

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

And a bit more:

$$\forall x\big(\exists y(Fxy \land \exists z(Dz \land Oyz)) \rightarrow \exists y(Dy \land Oxy \land \exists z(Fzx \land Ozy))\big).$$

And we are done!

17.4 Sentence Structure and Levels of Analysis

As I emphasised in §4.3, a single English sentence has many structures, depending on how fine-grained one is in the analysis of that sentence. We could symbolise 'Antony owns a car' in a number of different ways given the resources we have so far, in increasingly finer detail:

1. We could symbolise it as an atomic sentence of Sentential like 'A', ignoring all internal structure of the sentence (because it has no internal truth-functional structure). This is also a zero-place predicate, so is also a sentence of Quantifier.

2. We could symbolise it as another atomic sentence of Quantifier, but one which does recognise the internal subject-predicate struture of the English sentence. For example, we could symbolise it as 'Wa', where 'W' symbolises '$\underbrace{____}_{1}$ owns a car' and 'a' symbolises 'Antony'.

3. Or we could symbolise it as a complex quantified sentence '$\exists y(Cy \wedge Oay)$', where 'a' is as before, but 'C' means '$\underbrace{____}_{1}$ is a car' and 'O' is '$\underbrace{____}_{1}$ owns $\underbrace{____}_{2}$'.

4. We could imaginably symbolise it at even finer levels of structure, breaking down the predicate 'is a car' into the verb phrase 'is' and the indefinite noun phrase 'a car' (which is itself complex). This would go beyond the representational resources even of Quantifier.

Note that any structure identified is preserved: the name 'a' continues to appear in more fine-grained symbolisations once it has appeared. A one-place predicate, having appeared in the second symbolisation, still appears indirectly in the third symbolisation. For the open sentence '$\exists y(Cy \wedge Oxy)$' can be understood a representing a complex one-place predicate: 'x' is not associated with any quantifier, and can be replaced by a name to form a grammatical sentence.

How to symbolise the structure of a sentence very much depends on what purpose you have in symbolising. What matters is that you manage to represent enough structure to determine whether the target argument you are symbolising is valid. Valid arguments can be symbolised as invalid arguments, if you don't attend to the relevant structure, or don't have resources in your language to represent that structure. But we just observed that any more fine-grained analysis of the structure of a sentence retains the coarser structure (as it just adds more detailed substructure). So if you can show an argument is valid (conclusive in virtue of its structure) at some level of analysis, it will remain valid according to any more fine-grained understanding of the structure of that sentence. So you should aim to symbolise *just enough* structure in an argument to be able to demonstrate its validity – if indeed it is valid.

Key Ideas in §17

- The true power of Quantifier comes from its ability to handle multiple generality.
- But with great power comes additional complexity: we need to keep track of different places in our predicates, and different quantifiers that may govern those places. Even small alterations in scope or order can drastically change the meaning of the symbolisation we produce.
- Use of paraphrases and hybrid English-Quantifier sentences can be useful in figuring out how to symbolise a complex claim featuring multiple quantifiers and complex many-place predicates.

Practice exercises

A. Using this symbolisation key:

domain: all animals
A: $\underset{1}{\underline{\quad\quad}}$ is an alligator
M: $\underset{1}{\underline{\quad\quad}}$ is a monkey
R: $\underset{1}{\underline{\quad\quad}}$ is a reptile
Z: $\underset{1}{\underline{\quad\quad}}$ lives at the zoo
L: $\underset{1}{\underline{\quad\quad}}$ loves $\underset{2}{\underline{\quad\quad}}$
a: Amos
b: Bouncer
c: Cleo

symbolise each of the following sentences in Quantifier:

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
D: $\underset{1}{\underline{\quad\quad}}$ is a dog
S: $\underset{1}{\underline{\quad\quad}}$ likes samurai movies
L: $\underset{1}{\underline{\quad\quad}}$ is larger than $\underset{2}{\underline{\quad\quad}}$
b: Bertie
e: Emerson
f: Fergus

symbolise the following sentences in Quantifier:

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

C. Using this symbolisation key,

domain: all animals

L: $\underbrace{____}_{1}$ is larger than $\underbrace{____}_{2}$.

F: $\underbrace{____}_{1}$ is friendlier than $\underbrace{____}_{2}$.

D: $\underbrace{____}_{1}$ is a dog.

b: Bertie.

e: Emerson.

f: Fergus.

render the following into natural English, commenting on any difficulties:

1. $(Leb \land Lfe)$;
2. $\exists x(Dx \land Lxe)$;
3. $\big(\exists x(Dx \land Lxf) \to \exists y(Dy \land Fye)\big)$;
4. $\forall x(Dx \to \neg Lxf)$;
5. $\exists y((Lyb \land Ley) \lor (Lye \land Lby))$;
6. $\forall x(Dx \to \exists y(Dy \land Fxy))$;
7. $\forall x \forall y(Lxy \to \exists z(Dz \land (Fyz \land Fzx)))$.

D. Using the following symbolisation key:

domain: people and dishes at a potluck
R: $\underset{1}{\underline{\quad}}$ has run out.
T: $\underset{1}{\underline{\quad}}$ is on the table.
F: $\underset{1}{\underline{\quad}}$ is food.
P: $\underset{1}{\underline{\quad}}$ is a person.
G: $\underset{1}{\underline{\quad}}$ is guacamole.
L: $\underset{1}{\underline{\quad}}$ likes $\underset{2}{\underline{\quad}}$.
e: Eli
f: Francesca

render the following into natural English, commenting on any difficulties:

1. $\forall x(Fx \to Tx)$;
2. $(\exists x Gx \to \forall x(Gx \to Tx))$;
3. $\forall x \forall y((Px \land Gy) \to Lxy)$;
4. $\exists x \exists y(((Gx \land Py) \land Lyx) \to Lex)$;
5. $\forall x(Lfx \to Rx)$;
6. $\forall x(Px \to \neg(Lfx \lor Lxf))$;
7. $\forall x(\exists y(Gy \land Lxy) \to Lex)$;
8. $\forall x \forall y((Px \land Py) \to ((Lex \land Lyx) \to)Ley)$;
9. $(\exists x(Px \land Tx) \to \forall y(Fy \to Ry))$.

E. Using the following symbolisation key:

domain: people
D: $\underset{1}{\underline{\quad}}$ dances ballet.
F: $\underset{1}{\underline{\quad}}$ is female.
M: $\underset{1}{\underline{\quad}}$ is male.
C: $\underset{1}{\underline{\quad}}$ is a child of $\underset{2}{\underline{\quad}}$.
S: $\underset{1}{\underline{\quad}}$ is a sibling of $\underset{2}{\underline{\quad}}$.
e: Elmer
j: Jane
p: Patrick

symbolise the following arguments in Quantifier:

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

F. Consider the following symbolisation key:

domain: Fred, Amy, Kuiping
P: $\underbrace{____}_{1}$ pats $\underbrace{____}_{2}$ on the head.
T: $\underbrace{____}_{1}$ is taller than $\underbrace{____}_{2}$.

If we hold fixed this assignment of meanings to the predicates, why is it possible that '$\exists x \forall y Pxy$' is true, but not possible that '$\exists x \forall y Txy$' is true?

18

Identity

18.1 A Tricky Argument

Consider this sentence:

142. Pavel owes money to everyone.

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

O: $\underbrace{____}_{1}$ owes money to $\underbrace{____}_{2}$
p: Pavel

we can symbolise sentence 142 by '$\forall x Opx$'. But this has a (perhaps) odd consequence. It requires that Pavel owes money to every member of the domain (whatever the domain may be). The domain certainly includes Pavel. So this entails that Pavel owes money to himself.

Perhaps we meant to say:

143. Pavel owes money to everyone *else*
144. Pavel owes money to everyone *other than* Pavel
145. Pavel owes money to everyone *except* Pavel himself

We want to add something to the symbolisation of 142 to handle these italicised words. Some interesting issues arise as we do so.

18.2 First Attempt to Handle the Argument

The sentences in 143–145 can all be paraphrased as follows:

146. Everyone who isn't Pavel is such that: Pavel owes money to them.

This is a sentence of the form 'every F [person who is not Pavel] is G [owed money by Pavel]'. Accordingly it can be symbolised by something with this structure: $\forall x(\mathcal{A}x \rightarrow \mathcal{G}x)$. Here is an attempt to fill in the schematic letters:

O: $\underbrace{____}_{1}$ owes money to $\underbrace{____}_{2}$
I: $\underbrace{____}_{1}$ is $\underbrace{____}_{2}$
p: Pavel

With this symbolisation key, here is a proposed symbolisation: $\forall y(\neg Ipy \rightarrow Opy)$.

This symbolisation works well. But, it turns out, it doesn't quite do what we wanted it to. Suppose 'h' names Hikaru, and consider this argument, with the symbolisation next to the English sentences:

147. Pavel owes money to everyone else: $\forall y(\neg Ipy \rightarrow Opy)$.
148. Hikaru isn't Pavel: $\neg Ihp$.
So: Pavel owes money to Hikaru: Oph.

This argument is valid in English. But its symbolisation is not valid. If we pick Hikaru to be the value of 'y', we get from 147 the conditional $\neg Iph \rightarrow Oph$. But 148 doesn't give us the antecedent of this conditional: $\neg Iph$ is potentially quite different from $\neg Ihp$. So the argument isn't formally valid.

The argument isn't formally valid, because the sentence '$\neg Ihp$' doesn't formally entail '$\neg Ihp$' as a matter of logical structure alone. If the original argument is valid, then we need a symbolisation that as *a matter of logic* allows the distinctness (non-identity) of Hikaru and Pavel to entail the distinctness of Pavel and Hikaru.

18.3 Adding Identity

Logicians resolve this issue by adding identity as a new *logical* predicate – one of the structural words with a fixed interpretation.[1] We add a new symbol '=', to clearly differentiate it from our existing predicates.

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 fixed meaning. Like the quantifiers and connectives, the identity predicate does not need a symbolisation key to fix how it is to be used. Rather, it always gets the same interpretation: '=' always means '$\underbrace{____}_{1}$ is identical to $\underbrace{____}_{2}$'.

So identity is a special predicate because it has its meaning as part of logic.

[1] We don't absolutely have to do this: there are logical languages in which identity is not a logical predicate, and is symbolised by just choosing a two-place predicate like Ixy. But in our logical language Quantifier, we are choosing to treat identity as a structural word.

That one thing is logically identical to another does not mean *merely* that the objects in question are indistinguishable, or that all of the same things are true of them. When two things are alike in every respect, we may say they are QUALITATIVELY IDENTICAL. This is the sense of identity involved in 'identical twins', who are two distinct individuals who share their properties. In Quantifier, the identity predicate represents not this relation of similarity, but a relation of absolute or NUMERICAL IDENTITY: there is only one, rather than two. This is the sense in which Lewis Carroll (the author of *Alice in Wonderland*) is identical to Charles Lutwidge Dodgson (the Oxford mathematician): 'they' are the very same person, with two different names.

This might seem odd. Identity is a relation, but it doesn't relate different things to each other: it relates everything to itself, and to nothing else. We need a predicate for that relation because the names and (especially) variables of Quantifier aren't guaranteed to have different referents, and sometimes we want to explicitly require that two terms don't denote the same thing. For example, suppose we want to symbolise 'Barry is the tallest person'. You might try 'Barry is taller than everyone'. However, that would lead to the absurdity that Barry is taller than himself, since he is surely among 'everyone'. So what we really need is 'Barry is taller than everyone *else*, i.e., everyone who's not (identical to) Barry', which is most naturally formulated using the identity predicate.

18.4 Symbolising Identity Sentences

Now suppose we want to symbolise this sentence:

149. Pavel is Mister Checkov.

Let us add to our symbolisation key:

c: Mister Checkov

Now sentence 149 can be symbolised as '$p = c$'. This means that p is c, and it follows that the thing named by 'p' is the thing named by 'c'.[2]

Let's return to our example 'Barry is taller than everyone else'. We want to start with a paraphrase, like this: *choose anyone from the domain; if they are not Barry, than Barry is taller than them*. Where b symbolises 'Barry' and T symbolises '$\underset{1}{\underline{\quad}}$ is taller than $\underset{2}{\underline{\quad}}$', we might symbolise this as: '$\forall x(\neg x = b \rightarrow Tbx)$' (on the domain of people).

Using that same kind of structure, we can also now deal with sentences 143–145. All of these sentences can be paraphrased as 'Everyone who isn't Pavel is such that: Pavel owes money to them'. Paraphrasing some more, we get: 'For all x, if x is not Pavel, then x is owed money by Pavel'. Now that we are armed with our new identity symbol, we can symbolise this as '$\forall x(\neg x = p \rightarrow Opx)$'.

[2] One must be careful: the sentence '$p = c$' is, on this symbolisation, about Pavel and Mister Checkov; it is not about 'Pavel' and 'Mister Checkov', which are obviously distinct expressions of English.

This last sentence contains the formula '$\neg x = p$'. 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, '$x = p$'. But if this is confusing, you may use the NON-IDENTITY PREDICATE '$\neq$'. This is an abbreviation, characterised as folllows:

> Any occurrence of the expression '$x \neq y$' in a sentence abbreviates the expression '$\neg x = y$', and either can substitute for the other in any Quantifier sentence.

I will use both expressions in what follows, but strictly speaking '$\neg a = b$' is the official version, and we allow a conventional abbreviation '$a \neq b$'.

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

150. No one besides Pavel owes money to Hikaru.
151. Only Pavel owes Hikaru money.

Sentence 150 can be paraphrased as, 'No one who is not Pavel owes money to Hikaru'. This can be symbolised by '$\neg\exists x(\neg x = p \wedge Oxh)$'. Equally, sentence 150 can be paraphrased as 'for all x, if x owes money to Hikaru, then x is Pavel'. Then it can be symbolised as '$\forall x(Oxh \rightarrow x = p)$'. Sentence 151 can be treated similarly.[3]

18.5 Principles of Identity

Return to our argument from premises 147 and 148. We now symbolise it: $\forall y(p \neq y \rightarrow Opy); h \neq p \therefore Oph$. This argument is valid. Given the fixed interpretation we have assigned to '=', it is not possible that a first thing be not identical to a second, while the second is identical to the first. So we can conclude, as a matter of logical alone, that $h \neq p$ is equivalent to $p \neq h$.

This argument rests on one of the logical principles governing identity: that if a is b, then b is a. This property is known as symmetry, and is one of a cluster of basic properties governing the structure of identity:

> - Identity is REFLEXIVE: everything is identical to itself. So for any meaningful name 'a', '$a = a$' is true.
> - Identity is SYMMETRIC: if $a = b$, then also $b = a$.
> - Identity is TRANSITIVE: if $a = b$, and $b = c$, then also $a = c$.

3 But there is one subtlety here. Do either sentence 150 or 151 entail that Pavel himself owes money to Hikaru?

These principles of identity can be expressed as sentences of Quantifier. In the definitions given above, the names involved are arbitrary. So we can in fact paraphrase the reflexivity of identity as saying that for any thing, it is self-identical. Symbolised in Quantifier, this is '$\forall x x = x$'. Likewise for the others:

- Symmetry: $\forall x \forall y (x = y \rightarrow y = x)$;
- Transitivity: $\forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$.

These principles can apply to other two-place predicates too. For example, the two-place English predicate '$\underset{1}{\underline{\quad\quad}}$ is taller than $\underset{2}{\underline{\quad\quad}}$' is also transitive, since if Albert is taller than Barbara, and Barbara is taller than Chloe, then Albert must be taller than Chloe too. But it is not reflexive or symmetric: Albert is not taller than himself, and if Albert is taller than Barbara, it cannot be also that Barbara is taller than Albert. We will return to this topic in §21.10.

A final principle about identity is Leibniz' Law, named after the philosopher and mathematician Gottfried Leibniz:

> If $x = y$ then for any property at all, x has it iff y has it too. That is: every instance of this schematic sentence of Quantifier, for any predicate $\mathcal{F}$ whatsoever, is true:
>
> $$\forall x \forall y \big(x = y \rightarrow (\mathcal{F}x \leftrightarrow \mathcal{F}y)\big).$$

Leibniz' Law certainly entails that identical things are indistinguishable, sharing every property in common. But as we have already noted, identity isn't merely indistinguishability. Two things might be indistinguishable, but if there are two, they are not strictly identical in the logical sense we are concerned with. Yet in many cases, even very similar things do turn out to have some distinguishing property. There is a significant philosophical controversy over whether there can be cases of mere numerical difference, i.e., of nonidentity without any qualitative dissimilarity.

18.6 There are at Least ...

So far an identity predicate might seem useful in a few cases, like pseudonyms, but a bit niche. In fact, adding it to our language gives us the ability to say lots of things we cannot hope to say without it. In particular, the identity predicate gives us the ability to *count* – to quantify our quantifiers, and say how many things there are of a particular kind. Indeed, it is tempting to argue that the concept of counting depends on the prior concept of numerical distinctness.

For example, consider these sentences:

152. There is at least one apple
153. There are at least two apples
154. There are at least three apples

We shall use the symbolisation key:

A: $\underbrace{____}_{1}$ is an apple

Sentence 152 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 tempting to also translate sentence 153 without identity. Yet consider the sentence '$\exists x \exists y(Ax \wedge Ay)$'. Roughly, this says that there is some apple x in the domain and some apple y in the domain. Since nothing precludes these from being one and the same apple, this would be true even if there were only one apple.[4] In order to make sure that we are dealing with *different* apples, we need an identity predicate. Sentence 153 needs to say that the two apples that exist are not identical, so it can be symbolised by '$\exists x \exists y(Ax \wedge Ay \wedge \neg x = y)$'.

Sentence 154 requires talking about three different apples. Now we need three existential quantifiers, and we need to make sure that each will pick out something different: '$\exists x \exists y \exists z(Ax \wedge Ay \wedge Az \wedge x \neq y \wedge y \neq z \wedge x \neq z)$'.

18.7 There are at Most ...

Now consider these sentences:

155. There is at most one apple.
156. There are at most two apples.

Sentence 155 can be paraphrased as, 'It is not the case that there are at least *two* apples'. This is just the negation of sentence 153:

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

But sentence 155 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

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

The two sentences will turn out to be logically equivalent.

In a similar way, sentence 156 can be approached in two equivalent ways. It can be paraphrased as, 'It is not the case that there are *three* or more distinct apples', so we can offer:

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

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:

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

4 Note that both $\exists x Ax$ and $\exists y Ay$ are true in a domain with only one apple: the use of different variables doesn't require that different apples are the values of those variables.

18.8 There are Exactly ...

Now we can symbolise 'there are at least n' and we can symbolise 'there are at most n'. Using them together, we can symbolise 'there are exactly n':

157. There is exactly one apple.
158. There are exactly two apples.
159. There are exactly three apples.

Sentence 157 can be paraphrased as, 'There is *at least* one apple and there is *at most* one apple'. This is just the conjunction of sentence 152 and sentence 155. 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 157 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 158 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:

160. There are exactly two things.
161. 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, which is to say, they apply trivially to every thing. So we can symbolise either sentence with either of the following:

$$\exists x \exists y \neg x = y \land \neg \exists x \exists y \exists z(\neg x = y \land \neg y = z \land \neg x = z); \text{ or}$$
$$\exists x \exists y\big(\neg x = y \land \forall z(x = z \lor y = z)\big).$$

Key Ideas in §18

- Identity ('=') is the one logical predicate in Quantifier, with a fixed interpretation.
- Identity satisfies a number of logical principles, the most important of which is Leibniz' Law, that when x is y, then x is $\mathcal{F}$ iff y is $\mathcal{F}$, for any predicate $\mathcal{F}$.
- Identity is crucial for symbolising numerical quantification: 'there are at least n $\mathcal{F}$s', 'there are at most n $\mathcal{F}$s' and 'there are exactly n $\mathcal{F}$s', because these all involve – tacitly – the notion of distinctness between things.

Practice exercises

A. Explain why:

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

B. Using the following symbolisation key:

domain: all animals
D: $\underline{\quad\quad}_1$ is a dog
L: $\underline{\quad\quad}_1$ is larger than $\underline{\quad\quad}_2$
F: $\underline{\quad\quad}_1$ is fierce
b: Bertie
e: Emerson
f: Fergus

symbolise the following sentences in Quantifier:

1. Bertie is larger than all the other dogs.
2. Bertie, Emerson, and Fergus are all different dogs.
3. Emerson is smaller than at least two dogs.
4. The largest dog is not fierce.
5. One fierce dog is the same size as another fierce dog.

C. Using the following symbolisation key:

domain: cards in a standard deck

B: $\underset{1}{\underline{\quad\quad}}$ is black.

C: $\underset{1}{\underline{\quad\quad}}$ is a club.

D: $\underset{1}{\underline{\quad\quad}}$ is a deuce.

J: $\underset{1}{\underline{\quad\quad}}$ is a jack.

M: $\underset{1}{\underline{\quad\quad}}$ is a man with an axe.

O: $\underset{1}{\underline{\quad\quad}}$ is one-eyed.

W: $\underset{1}{\underline{\quad\quad}}$ is wild.

symbolise each sentence in Quantifier:

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.

D. Using the following symbolisation key:

domain: animals in the world

B: $\underset{1}{\underline{\quad\quad}}$ is in Farmer Brown's field.

H: $\underset{1}{\underline{\quad\quad}}$ is a horse.

P: $\underset{1}{\underline{\quad\quad}}$ is a Pegasus.

W: $\underset{1}{\underline{\quad\quad}}$ has wings.

symbolise the following sentences in Quantifier:

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

E. Identity is a reflexive, symmetric, and transitive predicate. Can you give examples of English predicates which are

1. Reflexive and symmetric but not transitive;
2. Reflexive and transitive but not symmetric;
3. Symmetric and transitive but not reflexive?

19

Definite Descriptions

In Quantifier, names function rather like names in English. They are simply labels for the things they name, and may be attached arbitrarily, without any indication of the characteristics of what they name.[1]

But complex noun phrases can also be used to denote particular things in English (recall §15.2), and they do so not merely by acting as arbitrary labels, but often by *describing* the thing they refer to. Consider sentences like:

162. Nick is the traitor.
163. The traitor went to Cambridge.
164. The traitor is the deputy.
165. The traitor is the shortest person who went to Cambridge.

These underlined noun phrases headed by 'the' – 'the traitor', 'the deputy', 'the shortest person who went to Cambridge' – are known as DEFINITE DESCRIPTIONS. They are meant to pick out a *unique* object, by using a description which applies to that object and to no other (at least, to no other salient object). The class of possessive singular terms, such as 'Antony's eldest child' or 'Facebook's founder', might be subsumed into the class of definite descriptions. They can be paraphrased using definite descriptions: 'the eldest child of Antony' or 'the founder of Facebook'.

Definite descriptions must be contrasted with INDEFINITE DESCRIPTIONS, such as 'A traitor went to Cambridge', where no unique traitor is implied. Definite descriptions must also be contrasted with what we might call *descriptive names*, such as 'the Pacific Ocean'. While the Pacific Ocean is an ocean, it isn't reliably peaceful, and even when it is, it surely isn't the unique ocean that merits that description. These descriptive name uses might also be involved in cases of GENERIC 'the', such as in 'The whale is

[1] This is not strictly true: consider the name 'Fido' which is conventionally the name of a dog. But even here the name doesn't carry any information in itself about what it names – the fact that we use that as a name only for dogs allows someone who knows that to reasonably infer that Fido is a dog. But 'Fido is a dog' isn't a trivial truth, as it would be if somehow 'Fido' carried with it the information that it applies only to dogs.

a mammal'. Here there is no implication that some specific whale is under discussion, but rather that the species is mammalian. (So maybe 'the whale' is a complex name for the species.) In the generic use, 'the whale is a mammal' can be paraphrased 'whales are mammals'. But a genuine definite description, such as 'the Prime Minister is a Liberal' cannot be paraphrased as 'Prime Ministers are Liberals'. The question we face is: can we adequately symbolise definite descriptions in Quantifier?[2]

19.1 Treating Definite Descriptions as Terms

One option would be to introduce new names whenever we come across a definite description. This is not a great idea. We know that *the* traitor – whoever it is – is indeed *a* traitor. We want to preserve that information in our symbolisation. So the symbolisation of 'The traitor is a traitor' (or 'the traitor is traitorous') should be a logical truth. But if we symbolise 'the traitor' by a name a, the symbolisation will be Ta, which is not a logical truth.

A second option would be to introduce a *new* definite description operator, such as 'ι'. The idea would be to symbolise 'the F' as 'ιxFx'. This is taken to mean something like this 'the unique thing such that it is F', which obviously involves a definite description in its semantics. Expressions of the form $\iota x\mathcal{A}x$ would then behave, grammatically speaking, like names – they combine with predicates to form sentences. Suppose we follow this path. Start with the following symbolisation key:

domain: people
T: $\underset{1}{\underline{\quad}}$ is a traitor
D: $\underset{1}{\underline{\quad}}$ is a deputy
C: $\underset{1}{\underline{\quad}}$ went to Cambridge
S: $\underset{1}{\underline{\quad}}$ is shorter than $\underset{2}{\underline{\quad}}$
n: Nick

We could symbolise sentence 162 with '$\iota xTx = n$' ('the thing which is a traitor is identical to Nick'), sentence 163 with '$C\iota xTx$', sentence 164 with '$\iota xTx = \iota xDx$', and sentence 165 with '$\iota xTx = \iota x(Cx \land \forall y((Cy \land x \neq y) \to Sxy))$'. This last example may be a bit tricky to parse. In semi-formal English, it says (supposing a domain of persons): 'the unique person such that they are a traitor is identical with the unique person such that they went to Cambridge and they are shorter than anyone else who went to Cambridge'.

However, even adding this new symbol to our language doesn't quite help with our initial complaint, since it is not self-evident that the symbolisation of 'The traitor is

[2] There is another question that I don't address: can we come up with a good theory of the meaning of 'the' in English that unifies how it behaves in 'the whale is a mammal', 'the Pacific Ocean is stormy', and 'Ortcutt is the shortest spy'? That question is very hard. Our task is to offer a symbolisation, and as we've seen, a symbolisation needn't be a translation, but only needs to capture the relevant implications to be successful.

a traitor' as '$T\iota x Tx$' yields a logical truth. More seriously, the idea that all definite descriptions are to be treated as terms makes it more difficult to give a unified treatment of descriptions in predicate position. It would be desirable to give a unified treatment of 'Ortcutt is a short spy' and 'Ortcutt is the short spy'; but while the former might be symbolised as '$(So \wedge Po)$', using the predicative 'is', the latter would need to treated as '$o = \iota x(Sx \wedge Px)$', using the 'is' of identity.

More practically, it would be nice if we didn't have to add a new symbol to Quantifier. And indeed, we might be able to handle descriptions using what we already have.

19.2 Russell's Paraphrase

Bertrand Russell offered an influential account of definite descriptions that might serve our purposes. Very briefly put, he observed that, when we say 'the $\mathcal{F}$', where that phrase is a definite description, our aim is to pick out the *one and only* thing that is $\mathcal{F}$ (in the appropriate contextually selected domain). Our discussion of counting in §18 gives us an idea about how Russell proposed to handle sentences expressing that there is exactly one $\mathcal{F}$.

Thus Russell offered a systematic paraphrase of sentences involving definite descriptions along these lines:[3]

the $\mathcal{F}$ is $\mathcal{G}$ iff: there is at least one $\mathcal{F}$, *and*
there is at most one $\mathcal{F}$, *and*
every $\mathcal{F}$ is $\mathcal{G}$

Note a very important feature of this paraphrase: *'the' does not appear on the right-side of the equivalence.* This approach would allow us to paraphrase every sentence of the same form as the left hand side into a sentence of the same form as the right hand side, and thus 'paraphrase away' the definite description.

It is crucial to notice that we can handle each of the conjuncts on the right hand side of the equivalence in Quantifier, using our techniques for dealing with numerical quantification. We can deal with the three conjuncts on the right-hand side of Russell's paraphrase as follows:

$$\exists x\mathcal{F}x \wedge \forall x\forall y((\mathcal{F}x \wedge \mathcal{F}y) \rightarrow x = y) \wedge \forall x(\mathcal{F}x \rightarrow \mathcal{G}x)$$

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 $\mathcal{F}$, and that the last conjunct tells us that that object is $\mathcal{G}$. So, equivalently, we could offer this symbolisation of 'The $\mathcal{F}$ is $\mathcal{G}$':

$$\exists x\big(\mathcal{F}x \wedge \forall y(\mathcal{F}y \rightarrow x = y) \wedge \mathcal{G}x\big)$$

Using these sorts of techniques, we can now symbolise sentences 162–164 without using any new-fangled fancy operator, such as 'ι'.

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

- Sentence 162 is exactly like the examples we have just considered. So we would symbolise it by '$\exists x(Tx \land \forall y(Ty \to x = y) \land x = n)$'.
- Sentence 163 poses no problems either: '$\exists x(Tx \land \forall y(Ty \to x = y) \land Cx)$'.
- Sentence 164 is a little trickier, because it links two definite descriptions. But, deploying Russell's paraphrase, it can be paraphrased by 'something is such that: there is exactly one traitor and there is exactly one deputy and it is each of them'. So we can symbolise it by:

  $$\exists x\Big(\big(Tx \land \forall y(Ty \to x = y)\big) \land \big(Dx \land \forall z(Dz \to x = z)\big)\Big).$$

  Note that I have made sure that both uniqueness conditions are in the scope of the initial existential quantifier.

Thus, we can adequately symbolise sentences involving definite descriptions in Quantifier. Incidentally, Russell also offers an account of indefinite descriptions of the same general form, differing only in that he regards indefinite descriptions as lacking any connotation of uniqueness. (One of the exercises below, p. 170, deals with that account.)

Let us dispel a worry. It seems that I can say 'the table is brown' without implying that there is one and only one table in the universe. But doesn't Russell's paraphrase literally entail that there is only one table? Indeed it does – it entails that there is only one table *in the domain under discussion*. While sometimes we explicitly restrict the domain, usually we leave it to our background conversational presuppositions to do so (recall §15.6). If I can successfully say 'the table is brown' in a conversation with you, for example, some background restriction on the domain must be in place that we both tacitly accept. For example, it might be that the prior discussion has focussed on your dining room, and so the implicit domain is *things in that room*. In that case, 'the table is brown' is true just in case there is exactly one table in that domain, and it is brown.

19.3 The Structure of Definite Descriptions

Russell offers his theory of definite descriptions as part of a campaign to effect 'a reduction of all propositions in which denoting phrases occur to forms in which no such phrases occur' ('On Denoting', p. 482). (Russell thought there were paradoxes attendant to descriptions in English that could be avoided if we showed how they could be systematically eliminated.) We do not wish to attempt anything so radical as this kind of *reductive analysis* – we only wanted to show that there is a way of symbolising definite descriptions in Quantifier. That is, we only need to assume that Russell's proposal allows us to *model* definite descriptions in Quantifier, a language that lacks any native resources for expressing them. Officially, then, we will take no stand on whether Russell's analysis is correct. Officially, we only suggest that Russell's analysis is the best approach to symbolising English sentences involving definite descriptions in Quantifier.

Yet Russell's account has some nice features that predict and explain some otherwise puzzling features of English 'the', and many logicians have followed Russell in thinking that the Russellian account might provide an adequate semantics for English definite descriptions. So in this section and in §19.4, I cannot resist discussing some of the evidence for Russell's account of the English 'the', and some of the major puzzles for that account. *These two sections should be regarded as optional.*

Empty Definite Descriptions One of the nice features of Russell's paraphrase is that it allows us to handle *empty* definite descriptions neatly. France has no king at present. Now, if we were to introduce a name, 'k', to name the present King of France, then everything would go wrong: remember from §15 that a name must always pick out some object in the domain, and whatever actual domain we choose, it will contain no present King of France. So we are at a loss to understand 'the King of France is bald': does it even say anything, since its subject has no referent?

Russell's paraphrase neatly avoids this problem. Russell tells us to treat definite descriptions using predicates and quantifiers, instead of names. Since predicates can be empty (see §16), this means that no difficulty now arises when the definite description is empty. The sentence 'the present King of France is bald' is paraphrased as 'there exists exactly one present King of France', and so turns out to be easily understood, and in fact to be false.

Scope and Descriptions Indeed, Russell's paraphrase helpfully highlights two ways one can go wrong with definite descriptions. To adapt an example from Stephen Neale,[4] suppose I, Antony Eagle, claim:

166. I am grandfather to the present king of France.

Using the following symbolisation key:

a: Antony
K: $\underbrace{____}_{1}$ is a present king of France
G: $\underbrace{____}_{1}$ is a grandfather of $\underbrace{____}_{2}$

Sentence 166 would be symbolised by '$\exists x(\forall y(Ky \leftrightarrow x = y) \land Gax)$'. Now, suppose you don't think this sentence 166 is true. You might express your rejection by saying:

167. Antony isn't the grandfather of the present king of France.

But your denial is *ambiguous*. There are two available readings of 167, corresponding to these two different sentences:

168. There is no one who is both the present King of France and such that Antony is his grandfather.
169. There is a unique present King of France, but Antony is not his grandfather.

4 Neale (1990) *Descriptions*, Cambridge: MIT Press.

Sentence 168 might be paraphrased by 'It is not the case that: Antony is a grandfather of the present King of France'. It will then be symbolised by '$\neg\exists x\big(Kx \wedge \forall y(Ky \rightarrow x = y) \wedge Gax\big)$'. We might call this WIDE SCOPE negation, since the negation takes scope over the entire sentence. Note that this sentence is predicted to be true, because the embedded sentence contains an empty definite description.

Sentence 169 can be symbolised by '$\exists x(Kx \wedge \forall y(Ky \rightarrow x = y) \wedge \neg Gax)$. We might call this NARROW SCOPE negation, since the negation occurs within the scope of the definite description. Note that its truth would require that there be a present King of France, albeit one who is a grandchild of Antony; so this sentence, unlike 168, is predicted to be false.

These two disambiguations of your rejection 167 have different truth values, so don't mean the same thing. So there are two different reasons you could have for your rejection of my claim. Are you accepting that the definite description refers and denying what I said of the present king of France? Or are you denying that the definite description refers, rejecting a more basic assumption of what I said?

The basic point is that the Russellian paraphrase provides two places in the symbolisation of 167 for the negation of 'isn't' to fit: either taking scope over the whole sentence, or taking scope just over the predicate 'Gax'. We see evidence that there are these two options in the ambiguity of 167, and so we should opt for a semantics, like Russell's, which has the resources to handle these ambiguities – in this case, by positing a quantifier scope ambiguity like those we discussed in §17.2.

The term-forming operator approach to definite descriptions cannot handle this contrast. There is just one symbolisation of the negation of 166 available in this framework: '$\neg Ga\iota xKx$'. The original sentence is false, so this negation must be true. Since sentence 169 is false, this sentence does not express the inner negation of sentence 166. But there is no way to put the negation elsewhere that will express the same claim as 169. ('$Ga\iota x\neg Kx$' clearly doesn't do the job – it says there is just one unique thing which isn't the present king of France, and Antony is grandfather to it!) So the sentence 'Antony isn't grandfather to the present king of France' has only one correct symbolisation, and hence only one reading, if the operator approach to definite descriptions is correct. Since it is ambiguous, with multiple readings, the 'ι'-operator approach cannot be correct – it doesn't provide a complex enough grammar for definite description sentences to allow for these kinds of scope ambiguities.

19.4 The Adequacy of Russell's Paraphrase

The evidence from scope ambiguity we just considered is a substantial point in favour of Russell's paraphrase, at least compared to the operator approach. But how successful is Russell's paraphrase in general? There is a substantial philosophical literature around this issue, but I shall content myself with two observations.

Presupposition and Negation One worry focusses on Russell's treatment of empty definite descriptions. If there are no $\mathcal{F}$s, then on Russell's paraphrase, both 'the $\mathcal{F}$ is $\mathcal{G}$' and its narrow scope negation, 'the $\mathcal{F}$ is not-$\mathcal{G}$', are false. Strawson suggested that such

sentences should not be regarded as false, exactly.[5] Rather, they both seem to assume that 'the $\mathcal{F}$' refers, and since this assumption is incorrect, the sentences misfire in a way that should, Strawson thinks, make us regard as *neither* true *nor* false, but nevertheless still meaningful.

A SEMANTIC PRESUPPOSITION of a declarative sentence is something that must be *taken for granted* by anyone asserting the sentence, triggered or forced by the words involved.[6] A pretty reliable test for whether $\mathcal{P}$ is a semantic presupposition of a sentence $\mathcal{A}$ is whether $\mathcal{P}$ is a consequence of both $\mathcal{A}$ and $\neg\mathcal{A}$. Strawson elevates this test to a *definition* of semantic presupposition: presuppositions are entailments that persist when a sentence is embedded under negation. So 'John has stopped drinking' and its negation 'John hasn't stopped drinking' both entail in English that John used to drink, and hence 'John used to drink' is a semantic presupposition of 'John has stopped drinking'. Here the presupposition is triggered by the aspectual verb 'stopped'.

Strawson says that PRESUPPOSITION FAILURE occurs when the presupposition of a sentence is false. If John never used to drink, both 'John has stopped drinking' and 'John hasn't stopped drinking' misfire. Strawson, following Frege, suggests that in cases of presupposition failure, a sentence is neither true nor false.

In the case of definite descriptions, the Frege-Strawson view would say that 'the present King of France is bald' presupposes that there is a present King of France. Since that presupposition fails, the sentence is neither true nor false. This is contrary to Russell's position that the sentence is false.

With the notion of presupposition failure in hand, the Frege-Strawson theory seems to be able to address the scope evidence for Russell's account. For there is now a distinction between the denial involved in 168 and that involved in 169, even though the logical form of the denied sentence is '$Ga\iota xKx$'. The logical negation of that sentence is '$\neg Ga\iota xKx$', which shares the presupposition that there is a present King of France. But there is also another way of rejecting a sentence, one which targets not what was said by the sentence, but its presuppositions. This is sometimes called METALINGUISTIC NEGATION.[7] One way of identifying its presence is the use of focal stress, emphasising the word to be targeted, and accompanied by an gloss explaining the presupposition to be rejected, as in:

170. I don't *like* cricket, I love it!
171. John hasn't *stopped* drinking; he never even started!
172. Sarah is *not* the source of the leak; there was no leak!

The effects which Russell sees as the result of scope ambiguity are re-analysed as involving the contrast between ordinary negation in 169, and metalinguistic negation in 168. Crucially, on the Frege-Strawson view, the successful deployment of metalinguistic negation in cases like 172 renders the sentence 'Sarah is the source of the leak' neither true nor false.

5 P F Strawson (1950) 'On Referring', *Mind* **59** pp. 320–34.

6 See David I Beaver, Bart Geurts, and Kristie Denlinger (2021) 'Presupposition', in Edward N Zalta, ed., *The Stanford Encyclopedia of Philosophy* https://plato.stanford.edu/archives/spr2021/entries/presupposition/, esp. §2 and §6.

7 Larry Horn (1989) *A Natural History of Negation*, University of Chicago Press.

The phenomenon of metalinguistic negation is real. But if we agree with Frege and Strawson here on how to model it semantically, we shall need to revise our logic. For, in our logic, there are only two truth values (True and False), and every meaningful sentence is assigned exactly one of these truth values. Why? Suppose there is presupposition failure of 'John has stopped drinking', because John never drank. Then 'John has stopped drinking' can't be true since it entails something false, its presupposition 'John drank'. And 'John hasn't stopped drinking' can't be true, since it also entails that same falsehood. So neither can be true. Since one is the negation of the other, if either is false, the other is true. So neither can be false, either. So if there are nontrivial semantic presuppositions – semantic presuppositions that might be false – then we shall have to admit that some meaningful sentences with a false presupposition are neither true nor false. It remains an open question, admittedly, whether presuppositions in the ordinary and intuitive sense are really semantic presuppositions in *this* sense.

But there is room to disagree with Strawson. Strawson is appealing to some linguistic intuitions, but it is not clear that they are very robust. For example: isn't it just *false*, not 'gappy', that Antony is grandfather to the present King of France? (This is Neale's line.)

Misdescription Keith Donnellan raised a second sort of worry, which (very roughly) can be brought out by thinking about a case of mistaken identity.[8] Two men stand in the corner: a very tall man drinking what looks like a gin martini; and a very short man drinking what looks like a pint of water. Seeing them, Malika says:

173. The gin-drinker is very tall!

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

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

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

Again, one might wonder how clear our intuitions are on this case. We can all agree that Malika intended to pick out a particular man, and say something true of him (that he was tall). On Russell's paraphrase, she actually picked out a different man (the short one), and consequently said something false of him. But maybe advocates of Russell's paraphrase only need to explain *why* Malika's intentions were frustrated, and so why she said something false. This is easy enough to do: Malika said something false because she had false beliefs about the men's drinks; if Malika's beliefs about the drinks had been true, then she would have said something true.[9]

8 Keith Donnellan (1966) 'Reference and Definite Descriptions', *Philosophical Review* **77**, pp. 281–304.

9 Interested parties should read Saul Kripke (1977) '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.

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

Key Ideas in §19

- Definite descriptions like 'the inventor of the zipper' are singular terms in English, but behave rather unlike names – for example, while the inventor of the zipper must be an inventor, Julius needn't be – even if Julius is the name of the inventor of the zipper.
- Russell's insight was if definite descriptions denote by uniquely describing, then we can use the ability of Quantifier to symbolise sentences like 'there is exactly one $\mathcal{F}$ and it is $\mathcal{G}$' to represent definite descriptions.
- There is an ongoing debate in linguistics about whether Russell's account captures the meaning of natural language definite descriptions, but there is no question that his approach is the only viable way to represent English descriptive noun phrases in Quantifier.

Practice exercises

A. Using the following symbolisation key:

domain: people

K: $\underline{\quad}_1$ knows the combination to the safe.

S: $\underline{\quad}_1$ is a spy.

V: $\underline{\quad}_1$ is a vegetarian.

T: $\underline{\quad}_1$ trusts $\underline{\quad}_2$.

h: Hofthor

i: Ingmar

symbolise the following sentences in Quantifier:

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: animals
Cx: $\underbrace{_____}_{1}$ is a cat.
G: $\underbrace{_____}_{1}$ is grumpier than $\underbrace{_____}_{2}$.
f: Felix
g: Sylvester

symbolise each of the following in Quantifier:

1. Some animals are grumpier than cats.
2. Some cat is grumpier than every other cat.
3. Sylvester is the grumpiest cat.
4. Of all the cats, Felix is the least grumpy.

C. Using the following symbolisation key:

domain: cards in a standard deck
B: $\underbrace{_____}_{1}$ is black.
C: $\underbrace{_____}_{1}$ is a club.
D: $\underbrace{_____}_{1}$ is a deuce.
J: $\underbrace{_____}_{1}$ is a jack.
M: $\underbrace{_____}_{1}$ is a man with an axe.
O: $\underbrace{_____}_{1}$ is one-eyed.
W: $\underbrace{_____}_{1}$ is wild.

symbolise each sentence in Quantifier:

1. The deuce of clubs is a black card.
2. One-eyed jacks and the man with the axe are wild.
3. If the deuce of clubs is wild, then there is exactly one wild card.
4. The man with the axe is not a jack.
5. The deuce of clubs is not the man with the axe.

D. Using the following symbolisation key:

domain: animals in the world
B: $\underbrace{_____}_{1}$ is in Farmer Brown's field.
H: $\underbrace{_____}_{1}$ is a horse.
P: $\underbrace{_____}_{1}$ is a Pegasus.
W: $\underbrace{_____}_{1}$ has wings.

symbolise the following sentences in Quantifier:

1. The Pegasus is a winged horse.
2. The animal in Farmer Brown's field is not a horse.
3. The horse in Farmer Brown's field does not have wings.

E. In this section, I symbolised 'Nick is the traitor' by '$\exists x(Tx \wedge \forall y(Ty \rightarrow x = y) \wedge x = n)$'. Two equally good symbolisations would be:

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

Explain why these would be equally good symbolisations.

F. Candace returns to her parents' home to find the family dog with a bandaged nose. Her mother says 'the dog got into a fight with another dog', and this seems a perfectly appropriate thing to say in the circumstances.

Does this example pose a problem for Russell's approach to definite descriptions?

G. Some people have argued that the following two sentences are ambiguous. Are they? If they are, explain how this fact might be used to provide support to Russell's paraphrases of definite description sentences.

1. The prime minister has always been Australian.
2. The number of planets is necessarily eight.

H. Russell's paraphrase of an indefinite description sentence like 'José met a man' is: *there is at least one thing x such that x is male and human and and José met x*.

Note that the word 'is' has apparently two readings: sometimes, as in 'Fido is heavy', it indicates *predication*; sometimes, as in 'Fido is Rover', it indicates identity (in the case, the same dog is known by two names). Something interesting arises if Russell's account of indefinite descriptions is right, since in an example like 'Fido is a dog of unusual size', we might interpret the 'is' in either way, roughly:

1. Fido is identical to a dog of unusual size;
2. Fido has the property of being a dog of unusual size.

Suppose that 'U' symbolises the property of being a dog of unusual size, then our two readings can be symbolised '$\exists x(Ux \wedge f = x)$' and '$Uf$'.

Is there any significant difference in meaning between these two symbolisations?

20

Sentences of Quantifier

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

20.1 Expressions

There are six kinds of symbols in Quantifier:

Predicate symbols	$A, B, C, \ldots, Z$
with subscripts, as needed	$A_1, B_1, Z_1, A_2, A_{25}, J_{375}, \ldots$
and the identity symbol	$=$.
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, of two types	
Truth-functional Connectives	$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
Quantifiers	$\forall, \exists$
Parentheses	(,)

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

20.2 Terms and Formulae

In §6, we went straight from the statement of the vocabulary of Sentential to the definition of a sentence of Sentential. In Quantifier, 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 FORMULAE.

> 1. If $\mathcal{R}$ is any predicate other than the identity predicate '=', and we have 0 or more terms $t_1, t_2, ..., t_n$ (not necessarily distinct from one another), then $\mathcal{R}t_1 t_2 ... t_n$ is an atomic formula.
> 2. If t_1 and t_2 are terms, then $t_1 = t_2$ is an atomic formula.
> 3. Nothing else is an atomic formula.

The use of script fonts here follows the conventions laid down in §7. So, '$\mathcal{R}$' is not itself a predicate of Quantifier. Rather, it is a symbol of our metalanguage (augmented English) that we use to talk about any predicate of Quantifier. Similarly, 't_1' is not a term of Quantifier, but a symbol of the metalanguage that we can use to talk about any term of Quantifier. So here are some atomic formulae:

$$x = a$$
$$a = b$$
$$Fx$$
$$Fa$$
$$Gxay$$
$$Gaaa$$
$$Sx_1 x_2 abyx_1$$
$$Sby_{254} zaaz$$

Remember that we allow zero-place predicates too, to ensure that sentence letters of Sentential are grammatical expressions of Quantifier too. According to the definition, any predicate symbol followed by no terms at all is also an atomic formula of Quantifier. So 'Q' by itself is an acceptable atomic formula.

Earlier, we distinguished many-place from one-place predicates. We made no distinction however in our list of acceptable symbols between predicate symbols with different numbers of places. This means that 'F', 'Fa', 'Fax', and '$Faxb$' are all atomic formulae. We will not introduce any device for explicitly indicating what number of places a predicate has. Rather, we will assume that in every atomic formula of the

form $\mathcal{A}t_1 \ldots t_n$, $\mathcal{A}$ denotes an n-place predicate. This means there is a potential for confusion in practice, if someone chooses to symbolise an argument using both the one-place predicate 'A' and the two-place predicate 'A'. Rather than forbid this entirely, we recommend choosing distinct symbols for different place predicates in any symbolisation you construct.

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

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 formulae, then $(\mathcal{A} \wedge \mathcal{B})$ is a formula.
4. If $\mathcal{A}$ and $\mathcal{B}$ are formulae, then $(\mathcal{A} \vee \mathcal{B})$ is a formula.
5. If $\mathcal{A}$ and $\mathcal{B}$ are formulae, then $(\mathcal{A} \rightarrow \mathcal{B})$ is a formula.
6. If $\mathcal{A}$ and $\mathcal{B}$ are formulae, then $(\mathcal{A} \leftrightarrow \mathcal{B})$ is a formula.
7. If $\mathcal{A}$ is a formula and x is a variable, then $\forall x\mathcal{A}$ is a formula.
8. If $\mathcal{A}$ is a formula and x is a variable, then $\exists x\mathcal{A}$ is a formula.
9. Nothing else is a formula.

Here are some formulae:

$$Fx$$
$$Gayz$$
$$Syzyayx$$
$$(Gayz \rightarrow Syzyayx)$$
$$\forall z(Gayz \rightarrow Syzyayx)$$
$$Fx \leftrightarrow \forall z(Gayz \rightarrow Syzyayx)$$
$$\exists y(Fx \leftrightarrow \forall z(Gayz \rightarrow Syzyayx))$$
$$\forall x\exists y(Fx \leftrightarrow \forall z(Gayz \rightarrow Syzyayx))$$
$$\forall x(Fx \rightarrow \exists xGx)$$
$$\forall y(Fa \rightarrow Fa)$$

We can now give a formal definition of scope, which incorporates the definition of the scope of a quantifier. Here we follow the case of Sentential, though we note that a connective can be either a connective or a quantifier:

The MAIN CONNECTIVE in a formula is the connective that was introduced last, when that formula was constructed using the recursion rules.

The SCOPE of a connective in a formula is the subformula for which that connective is the main connective.

So we can graphically illustrate the scope of the quantifiers in the last two examples thus:

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

Note that it follows from our recursive definition that '$\forall xFa$' is a formula. While puzzling on its face, with that quantifier governing no variable in its scope, it nevertheless is a formula of the language. It will turn out that, because the quantifier binds no variable in its scope, it is redundant; the formula '$\forall xFa$' is logically equivalent to 'Fa'. Eliminating such formulae from the language involves greatly increased complication in the definition of a formula for no important gain.

20.3 Sentences

Recall that we are largely concerned in logic with assertoric sentences: sentences that can be either true or false. Many formulae are not sentences. Consider the following symbolisation key:

domain: people
L: $\underbrace{____}_{1}$ loves $\underbrace{____}_{2}$
b: Boris

Consider the atomic formula 'Lzz'. All atomic formulae are formulae, 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 herself, in the same way that 'Lbb' is true just in case Boris (the person named by 'b') loves himself. *But 'z' is a variable, and does not name anyone or any thing.* It is true that we can sometimes manage to make a claim by saying 'it loves it', the best Engish rendering of 'Lzz'. But we can only do so by making use of contextual cues to supply referents for the pronouns 'it' – contextual cues that artifical languages like Quantifier lack. (If you and I are both looking at a bee sucking nectar on a flower, we might say 'it loves it' to express the claim that the bee [*it*] loves the nectar [*it*]. But we don't have such a rich environment to appeal to when trying to interpret formulae of Quantifier.)

Of course, if we put an existential quantifier out front, obtaining '$\exists zLzz$', then this would be true iff someone loves themself (i.e., someone [z] is such that they [z] love themself [z]). Equally, if we wrote '$\forall zLzz$', this would be true iff everyone loves themselves. The point is that, in the absence of an explicit introduction or contextual cues, 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 x that is within the scope of either $\forall x$ or $\exists x$.
> A FREE VARIABLE is any variable that is not bound.

For example, consider the formula

$$\forall x(Ex \vee Dy) \rightarrow \exists z(Ex \rightarrow Lzx)$$

The scope of the universal quantifier '$\forall x$' is '$\forall x(Ex \vee Dy)$', so the first 'x' is bound by the universal quantifier. However, the second and third occurrence of 'x' are free. Equally, the 'y' is free. The scope of the existential quantifier '$\exists z$' is '$\exists z(Ex \rightarrow Lzx)$', so '$z$' is bound.

In our last example from the previous section, '$\forall x(Fx \rightarrow \exists xGx)$', the variable '$x$' in '$Gx$' is bound by the quantifier '$\exists x$', and so x is free in '$Fx \rightarrow \exists xGx$' only when it appears in 'Fx'. So while the scope of '$\forall x$' is the whole sentence, it nevertheless doesn't bind every variable in its scope – only those such that, were it absent, would be free. (So we might say an occurrence of a variable x is BOUND BY an occurrence of quantifier $\forall x/\exists x$ just in case it would have been free had that quantifier been omitted.)

Finally we can say the following.

> The SENTENCES of Quantifier are just exactly those formulae of Quantifier that contain no free variables.

Since an atomic formula formed by a zero-place predicate contains no terms at all, and hence cannot contain a variable, every such expression is a sentence – they are just the atomic sentences of Sentential. Any other formula which contains no variables, but only names, is also a sentence, as well as all those formulae which contain only bound variables.

Our definition of a formula allows for examples like '$\exists x\forall xFx$'. This is a sentence, since the variable in 'Fx' is in the scope of a quantifier attached to 'x'. But which one? It could make a difference whether the sentence is to be understood as saying everything is F, or something it. To resolve this issue, let us stipulate that a variable is bound by the quantifier which is the main connective of the smallest subformula in which the variable is bound. So in '$\exists x\forall xFx$', the variable is bound by the universal quantifier, because it was already bound in the subformula '$\forall xFx$'.

20.4 Parenthetical Conventions

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

First, we may omit the outermost parentheses of a formula, and we sometimes use variably sized parentheses to aid with readability.

Second, we may omit parentheses between each pair of conjuncts when writing long series of conjunctions.

Third, we may omit parentheses between each pair of disjuncts when writing long series of disjunctions.

Key Ideas in §20

- There is a precise recursive definition of the notion of a sentence in Quantifier, describing how they are built out of basic expressions.
- The distinction between a formula and a sentence is important; in a sentence, no variable occurs without an associated quantifier binding it. Sentences, unlike mere formulae, contain all the information needed to understand their variables, once interpreted.

Practice exercises

A. Identify which variables are bound and which are free. Are any of these expressions formulas of Quantifier? Are any of them sentences? Explain your answers.

1. $\forall x(Ax \lor (Cx \to Bx))$
2. $\exists x(Lxy \land \forall yLyx)$
3. $(\forall xAx \land Bx)$
4. $(\forall x(Ax \land Bx) \land \forall y(Cx \land Dy))$
5. $(\forall x(Ax \land Bx) \land \forall xy(Cx \land Dy))$
6. $(\forall x\exists y\big(Rxy \to (Jz \land Kx)\big) \lor Ryx)$
7. $(\forall x_1(Mx_2 \leftrightarrow Lx_2x_1) \land \exists x_2Lx_3x_2)$

B. Identify which of the following are (a) expressions of Quantifier; (b) formulae of Quantifier; and (c) sentences of Quantifier.

1. $\exists x(Fx \to \forall y(Fx \to \exists xGx))$;
2. $\exists \neg x(Fx \land Gx)$;
3. $\exists x(Gx \land Gxx)$;
4. $(Fx \to \forall x(Gx \land Fx))$;
5. $(\forall x(Gx \land Fx) \to Fx)$;
6. $(P \land (\forall x(Fx \lor P)))$;
7. $(P \land \forall x(Fx \lor P))$;
8. $\forall x(Px \land \exists y(x \neq y))$.

Chapter 5

Interpretations

21

Extensionality

Recall that Sentential is a truth-functional language. Its connectives are all truth-functional, and *all* that we can do with Sentential is key sentences to particular truth values. We can do this *directly*. For example, we might stipulate that the Sentential 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 §8 that this should be taken to mean:

› The Sentential 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 I emphasised is that Sentential cannot handle differences in meaning that go beyond mere differences in truth value.

21.1 Symbolising Versus Translating: Extensional Languages

Quantifier has some similar limitations. It gets beyond mere truth values, since it enables us to split up sentences into terms, predicates and quantifier expressions. This enables us to consider what is *true of* some particular object, or of some or all objects. But we can do no more than that.

When we provide a symbolisation key for some Quantifier predicates, such as:

C: $\underbrace{____}_{1}$ lectures on logic in Adelaide in Semester 2, 2018

we do not carry the *meaning* of the English predicate across into our Quantifier predicate. We are simply stipulating something like the following:

› '*C*' and '$\underset{1}{\underline{\quad\quad}}$ lectures on logic in Adelaide in Semester 2, 2018' are to be *true of* exactly the same things.

So, in particular:

› '*C*' is to be true of all and only those things which lecture on logic in Adelaide in Semester 2, 2018 (whatever those things might be).

This is an indirect stipulation. Alternatively we can stipulate predicate extensions directly. We can stipulate that '*C*' is to be true of Antony Eagle and Jon Opie, and only them. As it happens, this direct stipulation would have the same effect as the indirect stipulation. But note that the English predicates '_____ is Antony Eagle or Jon Opie' and '_____ lectures on logic in Adelaide in Semester 2, 2018' have very different meanings!

The point is that Quantifier does not give us any resources for dealing with nuances of meaning. When we interpret Quantifier, all we are considering is what the predicates are actually true of. For this reason, I say only that Quantifier sentences *symbolise* English sentences. It is doubtful that we are *translating* English into Quantifier, for translations should preserve meanings.

The extension of an English expression is just the things to which it actually applies. So the extension of a name is the thing actually named; and the extension of a predicate is just those things it actually covers. Our symbolisation keys can be understood as stipulating that the extension of an Quantifier expression is to be the same as the extension of some English expression.

English is not an extensional language. In English, there is a distinction between the extension of a term – or what it *denotes* – and what it *means*. While the predicate '_____ is a monotreme' and the predicate '_____ is either an echidna or a platypus' have the same extension, and apply to the same creatures, they are not synonymous. For one thing, you can replace an expression being used in a sentence by one of its synonyms, and the resulting sentence will mean the same thing as the original – and will have the same truth value. But while 'Necessarily, all monotremes are monotremes' is true, 'Necessarily, all monotremes are either echidnas or platypuses' is not true. There are extinct species of monotreme, such as *Steropodon*, that are neither. Since there were monotremes that are neither platypus nor echnidna, that must be possible. So despite having the same current extension, these expressions differ in meaning. We need more than just the extension of some English expressions to fix the truth value of an English sentence involving them. Another example might be provided by singular terms. 'Scott Morrison' and 'the prime minister' share their extension, but differ in meaning: 'Always, Scott Morrison will be prime minister' is false, but 'Always, the prime minister will be prime minister' is trivially true. This difference in meaning is particularly evident when they are embedded in more complex constructions.

In Quantifier, by contrast, the substitition of one name for another with the same extension will always yield a sentence with the same truth-value; likewise with the substitution of one predicate for another with the same extension. This is normally summed up by saying that Quantifier is an EXTENSIONAL LANGUAGE. An extensional language is one where *non-logical* expressions with the same extension can always be swapped for one another in a sentence without changing the truth value.[1]

We noted above that a name and a definite description might have very different meanings, despite having the same extension. Our treatment of definite descriptions (§19) allows us to preserve the extensionality of Quantifier while also preserving the key logical features of descriptions. The definite description in English is analysed as a quantifier expression in Quantifier, so there is no such singular term as a definite description in Quantifier to be assigned an extension at all. What are assigned extensions are predicates and names, and our approach to definite descriptions allows the extensions of those predicates and names to fix the truth value of any symbolisation of an English sentence involving definite descriptions. The logical properties of the symbolised sentence in Quantifier, however, are also determined by the quantifier structure, which isn't fixed by assigning an extension.

21.2 A Word on Domains and Extensions

We always start our symbolisation keys by specifying a domain, and with extensions in view we can see why this is important. If we are to interpret an expression in terms of what it applies to, then we are going to need to know what things are 'out there' for it to apply to. As we said before (§15.6), the only restriction we place on our domains is that they be nonempty collections of things. But every extension we assign will be drawn from the domain: every name will be a member of the domain, and every one-place predicate will be assigned some sub-collection ('subset') of the domain.

We can stipulate directly which things in our domain our 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:

David Cameron
the number π
every top-F key on every piano ever made

[1] What this shows, in passing, is that Quantifier lacks the resources to express things like '_____ has always been _____' or '_____ is necessarily _____', which would allow us to separate expressions with the same present extension.

Another construction with apparently similar effects is when a true identity, such as 'Lewis Carroll is Charles Lutwidge Dodgson', is embedded in a belief report such as 'AE believes that Lewis Carroll is Charles Lutwidge Dodgson'. The belief report appears to be false, if AE doesn't know that 'Lewis Carroll' is a pen name for the Oxford mathematician. But still, this seems hard to deny: 'AE believes that Lewis Carroll is Lewis Carroll'. Many have used this to argue that the meaning of a name in English is not just its extension. But this is actually a rather controversial case, unlike the example in the main text. Many philosophers think that the meaning of a name in English just is its extension. But let us be clear: no one generalises this to predicates. Everyone agrees that the meaning of an English predicate is not just its extension.

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' (see below, §21.7). As long as the extension assigned consists of elements of the domain, we've managed to come up with a acceptable interpretation, at least from a purely logical point of view. Armed with this interpretation of 'H', suppose I now add to my symbolisation key:

d: David Cameron
n: Julia Gillard
p: the number π

Then 'Hd' and 'Hp' will both be true, on this interpretation, but 'Hn' will be false, since Julia Gillard was not among the stipulated objects.

This process of explicit stipulation is just giving the extension of a predicate by a list of items falling under it. A more common way of identifying the extension of a predicate is to derive it from a classification rule that sorts everything into those items that fall under it and those that do not. Such a classification rule is often what people think of as the meaning of a predicate. But such a classification rule goes beyond the extension. In effect, our earlier symbolisation keys assign an extension by relying on our knowledge of the classification rule associated with English predicates. While the rule might determine the extension in a domain, it is not the extension: two different rules could come up with the same extension, as in the case of monotremes earlier.

The extension of a one-place predicate is just a set of things. A set is identified by its members, so that set will be the same in every domain which contains all of its members. But we can see a further difference between extensions and classification rules when we can consider applying the same rule in different domains. So the classification rule associated with the English predicate '$_____$ is a student' applies to many many people in the domain of all people. In the domain 'people in this class', it applies only to a select few. In fact, you can even use the classification rule '$_____$ is a student' to fix an extension where the domain doesn't include any students at all. In that case it will just yield an EMPTY EXTENSION – one containing nothing. Here are some other cases where the empty extension should be assigned to a predicate:

- '$\underset{1}{_____}$ is a round square' (in any domain), or
- '$\underset{1}{_____}$ is a horse' in the domain of sheep, or
- '$\underset{1}{_____}$ is divisible without remainder by $\underset{2}{_____}$' in the domain of prime numbers (where the number 1 isn't normally thought to be a prime).

The empty collection too is a legitimate extension, because the empty set is still something: it is still a sub-collection of the domain. However, our names must always be assigned some element of the domain. An 'empty' name would have no extension at all

– and in an extensional language it is hard to see how we can admit such expressions. So keep clear the distinction between assigning an empty collection as an extension, and assigning nothing at all.

There are also trivial cases of the opposite sort, where we assign the entire domain as the extension of a predicate:

- '$\underset{1}{\underline{\quad\quad}}$ is something' (in any domain), or
- '$\underset{1}{\underline{\quad\quad}}$ is a horse' in the domain of horses, or
- '$\underset{1}{\underline{\quad\quad}}$ is greater than or equal to $\underset{2}{\underline{\quad\quad}}$' in the domain consisting of just the number 1 (the extension is ⟨1, 1⟩, which includes every ordered pair in this domain); and

21.3 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: $\underset{1}{\underline{\quad\quad}}$ loves $\underset{2}{\underline{\quad\quad}}$

Given what I said above, this symbolisation key should be read as saying:

- '*L*' and '$\underset{1}{\underline{\quad\quad}}$ loves $\underset{2}{\underline{\quad\quad}}$' are to be true of exactly the same things in the domain

So, in particular:

- '*L*' is to be true of a and b (in that order) iff a loves b.

It is important that we insist upon the order here, since love – famously – is not always reciprocated. (Note that 'a' and 'b' here are symbols of English, and that they are being *used* to talk about particular things in the domain.)

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 ORDERED PAIRS of objects, which differ from two-membered collections in that the order of a pair is important. Thus we might stipulate that '*B*' is to be true of, and only of, the following pairs of objects:

⟨Lenin, Marx⟩
⟨Heidegger, Sartre⟩
⟨Sartre, Heidegger⟩

Here the angle brackets keep us informed concerning order – ⟨a, b⟩ is a different pair from ⟨b, a⟩ even though they correspond to the same collection of two things, a and b. Suppose I now add the following stipulations:

l: Lenin
m: Marx
h: Heidegger
s: Sartre

Then 'Blm' will be true, since ⟨Lenin, Marx⟩ was in my explicit list. But 'Bml' will be false, since ⟨Marx, Lenin⟩ was not in my list. However, both 'Bhs' and 'Bsh' will be true, since both ⟨Heidegger, Sartre⟩ and ⟨Sartre, Heidegger⟩ are in my explicit list.

It is perhaps worth being explicit that our sequences are ordered groups of things, not lists of words or names. (Unless our domain is the set of words or names!) The extension of a many-place predicate is no less a worldly collection than the extension of a name or a one-place predicate. The sequences that are members of that extension have things as their constituents, not names.

To make these ideas more precise, we would need to develop some *set theory*, the mathematical theory of collections. This would give you some tools for modelling extensions and ordered pairs (and ordered triples, etc.), as well as some ideas about the metaphysical status of collections and sets. Indeed, set theoretic tools lie at the heart of contemporary linguistic approaches to meaning in natural languages. However, we have neither the time nor the need to cover set theory in this book. I shall leave these notions at an imprecise level; the general idea will be clear enough, I hope.

21.4 Zero-place Predicates

- Consider these examples:
  1. *Coober Pedy is a harsh place. It has underground houses.*
  2. *Coober Pedy is a harsh place. It seldom rains there.*
- In 1, 'it' refers to Coober Pedy. Not so in 2: 'It' in that sentence is a textbfdummy pronoun (though 'there' does refer)!
- We took this example to suggest that the textbfbare report 'it is raining' involved no genuine singular term, and modelled by a textbfzero-place predicate like R.
- These don't really apply to anything; they are either true or false of the whole speech situation, rather than true or false of something in the domain.
- We treat them as having textbftruth values as their extension – just like the textbfvaluations of Sentential. (In fact we can consider the valuations of Sentential as assignments of truth values as the extensions of atomic sentences of that language.)

21.5 Semantics for Identity

Identity is a special predicate of Quantifier. We write it a bit differently than other two-place predicates: '$x = y$' instead of 'Ixy' (for example). More important, though, its meaning is fixed as to be genuine *identity*, once and for all. Given a domain, the extension of '=' comprises just those pairs consisting of any member of the domain and itself. If the domain is numbers, for example, the extension of '=' on this domain will be

$$\langle 1, 1\rangle, \langle 2, 2\rangle, \langle 3, 3\rangle, \ldots.$$

If two names a and b are assigned to the same object in a given symbolisation key, and thus have the same extension, $a = b$ will also be true on that symbolisation key. And since the two names have the same extension, and Quantifier is extensional, substituting one name for another will not change the truth value of any Quantifier sentence. So, in particular, if 'a' and 'b' name the same object, then all of the following will be true:

$$Aa \leftrightarrow Ab$$
$$Ba \leftrightarrow Bb$$
$$Raa \leftrightarrow Rbb$$
$$Raa \leftrightarrow Rab$$
$$Rca \leftrightarrow Rcb$$
$$\forall xRxa \leftrightarrow \forall xRxb$$

This fact is sometimes called the INDISCERNIBILITY OF IDENTICALS, or Leibniz' Law (which we already encountered in §18.5): if two names denote the same thing, then no claim will be true when formulated using one of those names but false when formulated using the other in its place. That is because, at the level of extensions, there really is just one thing that those claims are about; how could that one thing be discernible from itself?

The converse claim – the IDENTITY OF INDISCERNIBLES – is much more controversial. It is not defensible in Quantifier. Suppose two objects differ in some feature $\mathfrak{F}$, but our symbolisation key includes no predicate which denotes $\mathfrak{F}$. Then we might not have any sentence true of the one object and false of the other, even though they are *distinct* objects which have different properties. We can't tell them apart, not because they are really just one, but because we don't have the right words to talk about how they are different. Quantifier thus allows that exactly the same predicates might be true of two distinct objects, and thus they are distinct but indiscernible when restricted to the resources of our symbolisation.

Once we move beyond Quantifier and other extensional languages, even the indiscernibility of identicals is controversial. For example, consider cases of secret identities, like in Spider-Man. Suppose we consider the English predicate 'MJ believes _____ shoots webs'. In the story, MJ believes that Spider-Man shoots webs, but she does not believe that Peter Parker shoots webs. Even though Spider-Man is Peter Parker, substituting one name for the other in this predicate turns a true sentence into a false one. An

open sentence which doesn't always yield the truth value when we substitute names with the same referent is called OPAQUE. The word which is responsible for the opacity here is the attitude verb 'believes'. Intuitively, what matters for a belief ascription is not which things are identical, but which things the subject represents as identical. Other attitude verbs also lead to opacity: 'knows', 'wants', 'remembers'. A language which contains opaque constructions cannot be completely represented in an extensional language like Quantifier.

21.6 Interpretations

I defined a *valuation* in Sentential as any assignment of truth and falsity to atomic sentences. In Quantifier, I am going to define an INTERPRETATION as consisting of three things:

- the specification of a domain;
- for each name that we care to consider, an assignment of exactly one object within the domain as its extension;
- for each nonlogical n-place predicate $\mathcal{A}$ (i.e., any predicate other than '=') that we care to consider, a specification of its extension: what things (or pairs of things, or triples of thing, etc.) the predicate is to be true of – where all those things must be in the domain.

A zero-place predicate (what we called 'atomic sentences' in Sentential) has a truth value, either T or F, as its extension – thus the valuations of Sentential are intepretations of a very restricted part of Quantifier – those interpretations which only care to consider zero-place predicates.

The symbolisation keys that I considered in chapter 4 consequently give us one very convenient way to present an interpretation. We shall continue to use them throughout this chapter. It is important to note, however, that the symbolisation key tends to associate each predicate with an already understood predicate (in English), and then uses that predicate as a classification rule to determine the extension of the predicate. We can, and sometimes do, offer an interpretation by simply giving the extensions of the intepreted Quantifier expressions directly. We can simply list the items, pairs, triples, etc., in the extension assigned to a predicate. So the following two symbolisation keys give the same intepretation:

- domain: Days of the week
  B: $\underbrace{____}_{1}$ is the day before $\underbrace{____}_{2}$

- domain: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
  B: ⟨Sunday, Monday⟩, ⟨Monday, Tuesday⟩, ⟨Tuesday, Wednesday⟩, ⟨Wednesday, Thursday⟩, ⟨Thursday, Friday⟩, ⟨Friday, Saturday⟩, ⟨Saturday, Sunday⟩.

So far, the domains we have used are drawn from the actual world, and English predicates have been used to assign their actual extensions to Quantifier predicates. But a perfectly good interpretation might assign merely possible entities to the domain, and merely possible extensions to the predicates of Quantifier, or a mixture of the two.

Let's therefore make the assumption that interpretations are not restricted to actuality:

> Any collection of objects, actual or merely possible, can serve as the domain of an interpretation. Any collection of objects drawn from that domain can serve as the extension of a one-place predicate, any collection of pairs of such objects can serve as the extension of a two-place predicate, etc.

A challenging question, which we will not address, is what enables us to talk about and apparently make use of such merely possible things in our interpretations.

21.7 Predicates, Properties, and Relations

A feature that might be shared by a number of different things is called a PROPERTY. These can be physical features like the property *being red*, or the property *having a mass of 2kg*; but we can also consider other properties that have a more social or mental basis, such as *being a logician*, *being rich*, or *being annoying*. In many cases, a property will determine an extension, because we can think of the property as dividing the domain in two: into a collection of things in the domain having a certain feature, and those lacking it. As the extension of a one-place predicate is a collection of things, the collection of things with a certain feature can correspond to an extension of a predicate. In that case we might say that the predicate represents the property. So on the domain of people, the collection of people who have the property of being a logician is an extension, and we can offer an interpretation on which this extension is assigned to some one-place predicate.

We have to be careful about this though, for several reasons. The first is that a property is a way for something to be, independent of the domain in which it might be included. An extension by contrast is defined as a subset of the domain. So a property might determine an extension, given a domain, but we should not take it to *be* an extension. Likewise, we should not understand the symbolisation key as assigning a property as the *meaning* of a predicate. Recalling §21.6, we might understand a symbolisation key as associate a Quantifier predicate with a property as expressed by an English predicate. But this is just a means to assign an extension to the predicate, using the property as a classification of the domain.

Another reason to hesitate in taking extensions to be properties is that, depending on the domain, many distinct properties will have the same extension. Consider the ratites, the bird family including ostriches, emus, cassowaries, and kiwis, among others. All living ratites are unable to fly. So on the domain of ratites, the property *being a bird* determines the same extension as the property *being flightless*. Suppose some predicate '*F*' is assigned this extension. If an extension is a property, then we'd have to

say that extension is both *is a bird* and *is flightless* – but since those two are distinct, the extension cannot be both, and in fact it is clear that it cannot be either.[2]

A final reason for caution is that many legitimate extensions don't seem to correspond to genuine properties. It is hard to characterise the distinction between genuine properties and the others, but one idea is that genuine properties contribute to *resemblance*. If two things are both red, then they resemble one another in appearance. But any subset of a domain is an extension, and we can imagine just selecting things almost at random to form an extension, which can then be assigned to a predicate. But there need be nothing those things have in common just because they are in the same set. Suppose I toss a coin ten times, and determine an extension by selecting n from the domain of natural numbers if the n-th toss was heads. The extension I get is: $1, 3, 4, 5, 7$. But clearly nothing in this recipe means this extension corresponds to a genuine property. Nothing in Quantifier requires that predicates must have genuine properties determining their extensions.

Properties determine extensions for one-place predicates. The corresponding entity determining the extension of a many-place predicate is a RELATION. So the relation of *loving* on a domain determines an extension, a set of pairs from the domain standing in that relation. The same caveats apply as in the case of properties: we shouldn't take Quantifier predicates to have relations as their meaning, we shouldn't identify a relation with its extension, and we shouldn't take any extension to correspond to a genuine relation.

21.8 Representing one-place predicates: Euler diagrams

One way of presenting an interpretation that is often convenient is to present its extensions *diagrammatically*.

The definition of an interpretation in §21.6 entailed that a one-place predicate should have as its extension a collection of things drawn from the domain. That is, the extension of a predicate is a SUBSET of the domain. We can represent subsets graphically using a EULER DIAGRAM, without having to specify each of the items which is in the domain.

We begin by demarcating a region on the page to represent the domain – typically, a rectangle. Then for each one-place predicate, we can represent it by a sub-region of the domain, subject to the following rules:

- The interior of a region associated with predicate $\mathcal{P}$ represents the members of the domain that are in the extension of $\mathcal{P}$; the exterior – the region within the domain but outside the given region – represents those members of the domain that are not in the extension of $\mathcal{P}$;

[2] These properties are coextensive on this domain, but not in general. But some properties seem to be necessarily coextensive: take the properties *being a plane figure with three sides* and *being a plane figure with three vertices*. At first glance these are different properties – one involves counting lines, the other counting angles.

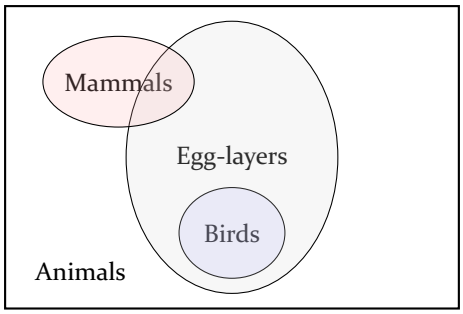


Figure 21.1: An interpretation represented diagrammatically.

- If two predicates have extensions that overlap, the regions that represent them in the diagram should overlap.
- If one predicate has an extension that is entirely within the extension of another predicate, the region associated with the first should be entirely within the region associated with the second;
- If two predicates have nonoverlapping extensions, the regions that represent them should not overlap.

We then label and shade each region to enable us to tell which predicates they are associated with. We may also add labelled dots to denote individuals who may fall into various predicate extensions.

These rules are quite abstract, but they are easy to apply in practice. Suppose we are given the following interpretation:

domain: animal species
E: $\underbrace{_____}_{1}$ lays eggs;
M: $\underbrace{_____}_{1}$ is a mammal;
B: $\underbrace{_____}_{1}$ is a bird;

We may represent this interpretation as in Figure 21.1. In this diagram, the egg-layers are a subset of the animals – not all animals reproduce by laying eggs, so they do not coincide with the whole domain. The birds are a subset of animals too, and in fact wholly within the egg-layers: all birds reproduce oviparously. The mammals are a subset of the animals, and one that overlaps with the egg-layers (the monotremes!), though neither is contained in the other, so the regions overlap without either being wholly within the other. The birds and mammals do not have any common members, so the associated regions do not overlap at all. Note that the sizes of the regions may, but need not, represent the number of things within them. In this diagram they probably don't.

Another example. Consider this interpretation, discussed earlier on page 127:

domain: people and plants
C: $\underbrace{____}_{1}$ is a cowboy
S: $\underbrace{____}_{1}$ sings a sad, sad song
R: $\underbrace{____}_{1}$ is a rose
T: $\underbrace{____}_{1}$ has a thorn

The wisdom of Bret Michaels then tells us that the Euler diagram should look like the representation in Figure 21.2.

We can start to see how interpretations can help us evaluate arguments quite vividly in this graphical environment. Recall this example from §15: 'Willard ('w') is a logician ('L'); Every logician wears a funny hat ('F') ∴ Willard wears a funny hat'. This can be represented using an Euler diagram making use of a labelled dot to represent the individual Willard, as in Figure 21.3. We can see from the diagram that this argument is valid: Willard falls in the region corresponding to 'F' because he falls in the region corresponding to 'L', which is wholly within the 'F'-region.

If you would like a real challenge, you might try to figure out the interpretation corresponding to the Euler diagram in Figure 21.4.

21.9 Representing two-place predicates: directed graphs

Suppose we want to consider just a single two-place predicate, 'R'. Then we can represent it, in some cases, by depicting the individual members of the domain by little dots (possibly with labels), and drawing a single-headed arrow from the first to the second of an ordered pair of items just in case that ordered pair falls within the extension of the predicate. Such a diagram is known as a DIRECTED GRAPH. A directed graph comprises a collection of NODES, and a collection of arrows (ordered links) between those nodes; in our case, the nodes are the elements of the domain, and the arrows correspond to the extension of a two-place predicate on that domain, given the convention

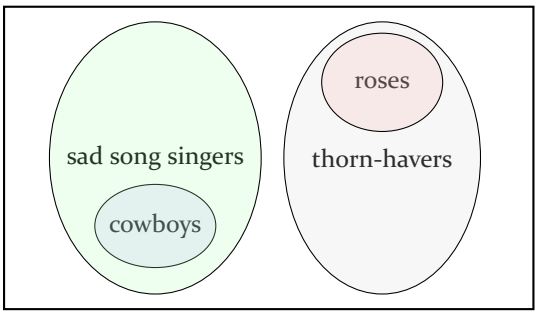


Figure 21.2: A representation of 'Every Rose Has Its Thorn'.

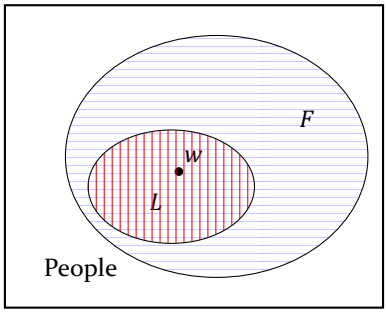


Figure 21.3: Representing the argument from §15.

that when there is an arrow running from x to y in a graph, that means $\langle x, y\rangle$ is in the extension of the predicate.

Let's consider some examples.

› First, consider the following interpretation, written in the more standard manner:

domain: $1, 2, 3, 4$
R: $\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle, \langle 4, 1\rangle, \langle 1, 3\rangle$.

That is, an interpretation whose domain is the first four positive whole numbers, and which interprets 'R' as being true of and only of the specified pairs of

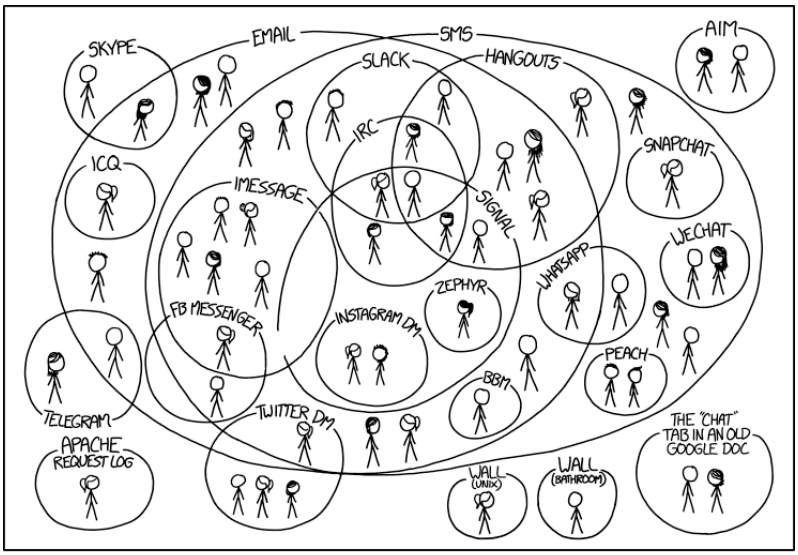


Figure 21.4: 'Chat Systems', https://xkcd.com/1810/

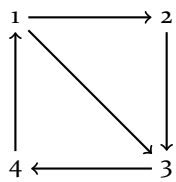


Figure 21.5: A simple graph.

numbers. This might be represented by the simple graph depicted in Figure 21.5.

› Consider the following interpretation:

domain: $1, 2, 3, 4$
R: $\langle 1, 3\rangle$, $\langle 3, 1\rangle$, $\langle 3, 4\rangle$, $\langle 1, 1\rangle$, $\langle 3, 3\rangle$, $\langle 4, 4\rangle$.

We might offer the graph in Figure 21.6 to represent it. The existence of pairs like $\langle 1, 1\rangle$ in the extension of R is borne out in the presence of 'loops' from nodes to themselves in the graph. (Because of these loops, our graph is not what graph theorists call a 'simple' graph.) Notice that 2 is in the domain, and hence included as a node in the graph, but it has no arrows attached to it, because it is not included in the extension assigned to R.

› A third example is depicted in Figure 21.7, corresponding to this interpretation:

Domain: Amia, Barbara, Corine, Davis
F: ⟨Amia, Amia⟩, ⟨Barbara, Barbara⟩, ⟨Davis, Davis⟩, ⟨Amia, Barbara⟩, ⟨Amia, Corine⟩, ⟨Barbara, Amia⟩, ⟨Barbara, Corine⟩, ⟨Corine, Barbara⟩

If we wanted, we can extend our graphical conventions, making our diagrams more complex in order to depict more complex interpretations. For example, we could introduce another kind of arrow (maybe with a dashed shaft) to represent a further two-place predicate.[3] We could add names as labels attached to particular objects. To

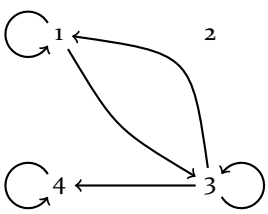


Figure 21.6: A graph with 'loops'

3 This kind of 'directed multigraph' is sometimes called a QUIVER.

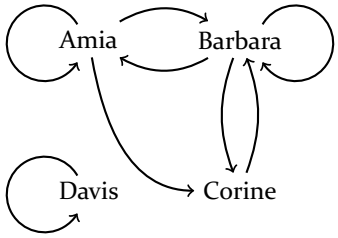


Figure 21.7: A more complicated graph.

symbolise the extension of a one-place predicate, we might simply adopt the approach from §21.8, and mark a shaded region around some particular objects and stipulate that the thus encircled objects (and only them) are to fall in the extension of some predicate 'H', say.[4]

All of these graphical innovations are used in Figure 21.8, which graphically represents the following interpretation:

domain: $2, 3, 4, 5, 6$

G: $\underset{1}{\underline{\quad}} \geqslant 4$.

D: $\underset{1}{\underline{\quad}}$ is distinct from and exactly divisible by $\underset{2}{\underline{\quad}}$;

T: $\underset{1}{\underline{\quad}} + 3 = \underset{2}{\underline{\quad}}$;

f: 4;

Here the label 'f' represents the name attached to 4, the blue dotted arrow represents the relation assigned to T, the black solid arrow represents the relation assigned to D, and the grey ellipse represents the collection of things in the domain which fall in the extension assigned to G.

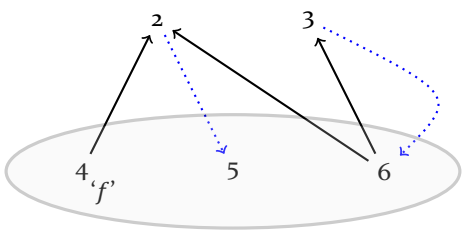


Figure 21.8: Multiple techniques used to depict a complex interpretation.

4 We needn't stop there. For example, we could introduce DIRECTED HYPERGRAPHS, where the edges can join more than two nodes.

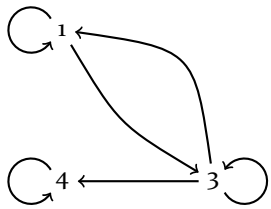


Figure 21.9: A graph of a reflexive relation on $\{1, 3, 4\}$.

21.10 Properties of Binary Relations

The extension of a two-place predicate is a collection of ordered pairs drawn from a domain D. Such a collection is also known as a BINARY RELATION on D. Binary relations have many interesting features. We have seen some of them already in §18.5, when we noted that reflexivity, symmetry and transitivity are all features of identity. We will now treat these notions more generally. Binary relations provide some good examples of how directed graphs can help grasp an interpretation, because the properties of binary relations often correspond to easily grasped conditions on the associated graphs.

› *This section should be regarded as optional, and only for the dedicated student of logic.*

› A binary relation $\mathfrak{R}$ on D is REFLEXIVE iff for any x in D, $\langle x,x \rangle$ is in $\mathfrak{R}$.

› $\mathfrak{R}$ is IRREFLEXIVE on D iff for no x in D is $\langle x, x \rangle$ in $\mathfrak{R}$.

› If $\mathfrak{R}$ is neither reflexive nor irreflexive, it is nonreflexive.

An example of a reflexive relation is the extension of '$\underline{\quad}_1$ is the same height as $\underline{\quad}_2$' on the domain of people. Everyone is the same height as themselves. An example of an irreflexive relation is the extension of '$\underline{\quad}_1$ is taller than $\underline{\quad}_2$' on the domain of people, since no one is taller than they are. An example of a nonreflexive relation might be the extension of '$\underline{\quad}_1$ trusts the judgment of $\underline{\quad}_2$' on the domain of people: some people trust their own judgement, while others second-guess themselves. Pictorially, a reflexive relation corresponds to a graph in which every node in the graph has an arrow pointing to itself, such as in Figure 21.9. Note the resemblance between this graph and the graph in Figure 21.6. They are almost the same, except the earlier Figure shows a relation in which the domain contains the number 2, which is *not* paired with itself in the relation. A relation being reflexive requires everything in the domain relate to itself: in Figure 21.6, that condition is not satisfied. The graph of an irreflexive relation has no loops from a node directly to itself, such as the relation depicted in Figure 21.5.

The graph of a nonreflexive relation will have a mixture of nodes with and without loops, such as in Figure 21.6.

Reflexivity is a property of an extension, not a predicate (except the identity predicate). But there is a sentence of Quantifier which expresses reflexivity. Suppose the two-place predicate 'R' is assigned a relation $\mathfrak{R}$, a set of pairs, as its extension on D: then '$\forall x Rxx$' ('everything bears R to itself') will be true under this interpretation iff $\mathfrak{R}$ is reflexive.

> - A binary relation $\mathfrak{R}$ is TRANSITIVE iff for any x, y, and z, whenever ⟨x, y⟩ and ⟨y, z⟩ are in $\mathfrak{R}$, then ⟨x, z⟩ is also in $\mathfrak{R}$.
> - $\mathfrak{R}$ is INTRANSITIVE iff for any x, y, and z, whenever ⟨x, y⟩ and ⟨y, z⟩ are in $\mathfrak{R}$, then ⟨x, z⟩ is *not* in $\mathfrak{R}$.
> - $\mathfrak{R}$ is nontransititive iff it is neither transitive nor intransitive.
>
> If the predicate 'R' is assigned $\mathfrak{R}$ as its extension in some interpretation, then $\mathfrak{R}$ is transitive iff '$\forall x \forall y \forall z((Rxy \land Ryz) \rightarrow Rxz)$' holds in this interpretation.

An example of a transitive relation is the extension of '$\underline{\quad}_1$ is older than $\underline{\quad}_2$' on the domain of buildings. Whenever one building is older than another, which is older than a third, the first must be older than the third. An example of an intransitive relation is the extension of '$\underline{\quad}_1$ is the successor of $\underline{\quad}_2$' on the domain of numbers, where x is the successor of y iff $x = y + 1$. While 7 is the successor of 6, and 6 is the successor of 5, it is not the case that 7 is the sucessor of 5. An example of a nontransitive relation is the extension of 'there is a direct flight between $\underline{\quad}_1$ and $\underline{\quad}_2$' on the domain of the Qantas network. There is a direct flight from Adelaide to Perth, and a direct flight from Perth to Broome, but no direct flight from Adelaide to Broome. On the other hand, there is a direct flight from Perth to Sydney, and there is a direct flight from Adelaide to Sydney.

Pictorially, a reflexive relation is one where there is a 'shortcut' between any two nodes that can be reached from one another by travelling along arrows in the intended direction. Consider the graph in Figure 21.10. Intuitively, one can 'get from' the oldest (the Mitchell Building) to the youngest (the Napier Building) via the intermediate aged Elder Hall and Bonython Hall. But you can also go directly, following the topmost curve. The graph of an intransitive relation has no shortcuts of this sort. The graph of a nontransitive relation has a mix of shortcuts, such as the graph of our Qantas example depicted in Figure 21.11.

One interesting question: is the relation depicted in Figure 21.6 intransitive? There are no obvious shortcuts; while you can get from 1 to 3, and from 3 to 4, you cannot go directly from 1 to 4. But those loops actually make for some degenerate shortcuts. E.g., there is an edge from 1 (x) to 3 (y), and from 3 (y) to 3 (z), and there is the 'shortcut' from 1 (x) to 3 (z). This may not be how you were thinking of transitivity, but look back

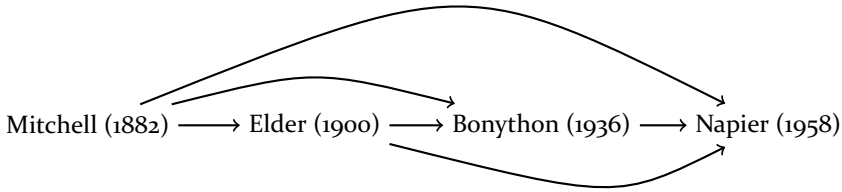


Figure 21.10: A graph of the transitive relation 'older than' on some University of Adelaide buildings.

at the definition, which is phrased in terms of picking *any* pairs from the domain. This even includes those pairs consisting of a node and itself.

- A binary relation $\mathfrak{R}$ is SYMMETRIC iff for any ⟨x, y⟩ in $\mathfrak{R}$, ⟨y, x⟩ is also in $\mathfrak{R}$.
- $\mathfrak{R}$ is ASYMMETRIC iff for any ⟨x, y⟩ in $\mathfrak{R}$, ⟨y, x⟩ is *not* in $\mathfrak{R}$.
- $\mathfrak{R}$ is ANTISYMMETRIC iff for any ⟨x, y⟩ in $\mathfrak{R}$, ⟨y, x⟩ is in $\mathfrak{R}$ *only if* $x = y$.
- $\mathfrak{R}$ is NONSYMMETRIC iff it is neither symmetric, asymmetric, nor antisymmetric.

If the predicate 'R' is assigned $\mathfrak{R}$ as its extension in some interpretation, then $\mathfrak{R}$ is symmetric iff '$\forall x \forall y (Rxy \rightarrow Ryx)$' holds in this interpretation.

An example of a symmetric relation is the extension of '$\underbrace{____}_{1}$ lives next door to to $\underbrace{____}_{2}$' on the domain of people. If Alf lives next to Beth, then Beth also lives next door

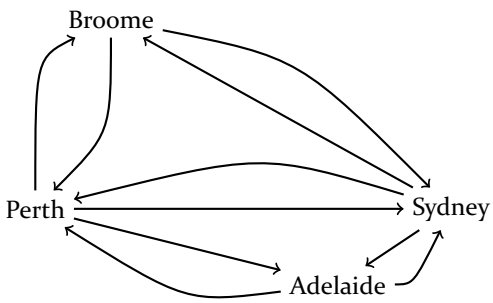


Figure 21.11: An extract of Qantas' route network.

to Alf. An asymmetric relation is '$\underset{1}{\underline{\quad\quad}}$ is greater than $\underset{2}{\underline{\quad\quad}}$' on the natural numbers: if $x > y$, then it cannot also be that $y > x$. If we consider not 'greater than', but the weaker relation 'greater than or equal to', we see an example of an antisymmetric relation: if $x \geq y$ then it can be the case that $y \geq x$ only in the special case where $x = y$. An example of a nonsymmetric relation might be the relation '$\underset{1}{\underline{\quad\quad}}$ loves $\underset{2}{\underline{\quad\quad}}$' on the domain of people: sometimes love is requited, so that both members of a given pair love each other; and sometimes it is unrequited.

Pictorially, whenever we have an arrow from one node to another, if there is also a 'reverse' arrow back from the second to the first, then the relation depicted is symmetric. See the depiction of the 'next to' relation in Figure 21.12. A relation is asymmetric if there are never such reverse arrows. A relation is antisymmetric if the only time there are arrows from x to y and back is when $x = y$ and there is a loop. I depict both relations $>$ and $\geq$ in Figure 21.13; the difference is that that the antisymmetric relation $\geq$ has an arrow from each node to itself.

The observant reader will have noticed that, among these definitions of properties of binary relations, the only definition which actually mentions the domain is the definition of reflexivity. All the other definitions are *conditional* in form: they say, if certain pairs are in the extension, then certain other pairs will (or won't) be too. We needn't specify a domain to check whether these conditionals hold of any relation. But to check whether a relation is reflexive, we need not only the ordered pairs of the relation, but also what the domain is, so we can see if any members of the domain are missing from the relation.[5]

These properties of binary relations are not completely independent of each other. For example, a reflexive relation cannot be asymmetric: a relation $\mathfrak{R}$ is asymmetric only if it is never the case that for any x and y, $\langle$x,y$\rangle$ is in $\mathfrak{R}$; but if the relation is reflexive, then every pair consisting of something and itself is in $\mathfrak{R}$, and nothing prohibits us from picking the same value for x and y. A reflexive relation can at best be antisymmetric.

A relation which is reflexive, symmetric and transitive is known as an EQUIVALENCE RELATION. We've already established that identity is an equivalence relation in §18.5. But

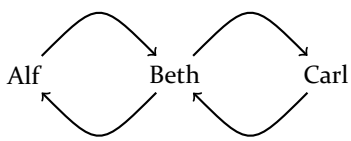


Figure 21.12: 'next to': a symmetric but intransitive relation.

5 Reflexivity is an *extrinsic* property of a relation – you can't tell just from the extension whether a relation is reflexive, because it is relative to the domain from which the relata of the relation may be drawn. The others are *intrinsic* properties of a relation; whether the relation has them is determined just by which pairs are in the relation. The very same set of pairs might be reflexive on one domain and nonreflexive on another, but it will be symmetric on every domain if it is symmetric on any.

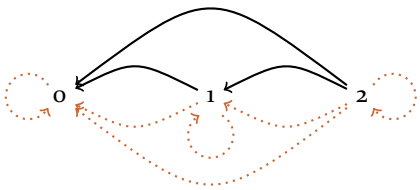


Figure 21.13: > (black arrows) and ≥ (orange dotted arrows) on the domain {0, 1, 2}.

there are other equivalence relations too: consider '$\underset{1}{\underline{\quad}}$ is the same height as $\underset{2}{\underline{\quad}}$'. This relation will structure the domain into clusters of people with the same height as each other. Such a division into groups is known as a PARTITION, and the individual groups are known as cells. When you partition a domain, you sort the domain into cells which are uniform with respect a given feature – in this case, height. There will be no connections between cells of the partition, but within each cell, each person will be related to every other person in the cell. These cells are also known as EQUIVALENCE CLASSES. Identity is the extreme case of an equivalence relation, because it partitions the domain into cells containing entities that are equivalent in every respect, i.e., are identical, and hence each cell contains just one individual member.

A relation on a domain D is TOTAL iff for any x and y in D, either ⟨x,y⟩ or ⟨y,x⟩ (or both) is in the extension.[6] Any two things are related, in some way, by a total relation. We can use the notion of totality to define a kind of relation that is of particular mathematical importance:

> A relation $\mathfrak{R}$ on a domain D is an ORDERING iff $\mathfrak{R}$ is reflexive, transitive, and antisymmetric. Special cases:
>
> - $\mathfrak{R}$ is a STRICT ORDER iff $\mathfrak{R}$ is an asymmetric order; i.e., iff $\mathfrak{R}$ is reflexive, transitive, and asymmetric.
> - $\mathfrak{R}$ is a TOTAL ORDER iff $\mathfrak{R}$ is total and is an order;
> - $\mathfrak{R}$ is a STRICT TOTAL ORDER iff $\mathfrak{R}$ is total and is a strict order.
>
> An order that is not total is commonly called a PARTIAL ORDER.

An order, as the name suggests, gives a particular kind of structure to a domain. A strict total order arranges everything in the domain in an order without any ties; an example would be the greater than relation > we considered above. A non-mathematical example might be '$\underset{1}{\underline{\quad}}$ is taller than $\underset{2}{\underline{\quad}}$', which arranges the domain of people in a linear way from tallest to shortest. If there are ties permitted, then the relation is not

6 If the predicate 'R' is assigned $\mathfrak{R}$ as its extension in some interpretation, then $\mathfrak{R}$ is total iff '$\forall x \forall y (Rxy \vee Ryx)$' holds in this interpretation.

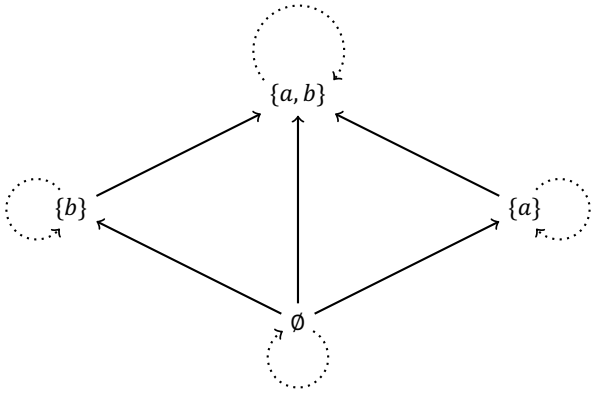


Figure 21.14: Black arrows indicate both $\subset$ and $\subseteq$, dotted arrows indicate $\subseteq$ only, on the domain $\wp\{a, b\} = \{\{a, b\}, \{a\}, \{b\}, \emptyset\}$.

strict: a total order with ties is $\geq$ on the natural numbers, or the relation '$\underset{1}{____}$ is no shorter than $\underset{2}{____}$', which allows that whatever fills the first slot might be either taller than, or the same height as, whatever fills the second.

If not everything in the domain is ordered with respect to each other, the relation is partial. Perhaps not everything in the domain is comparable. Suppose we are considering a railway network, in which all lines radiate from a central station, like the Adelaide Metro network. The relation '$\underset{1}{____}$ is no further along the line than $\underset{2}{____}$' is a partial order.[7] But there are many pairs of incomparable stations: Woodville is not on the same line as Oaklands, so neither is further along the line than the other. (We could create a total order, perhaps by measuring distance or counting stations along the line, and say that two stations are equally far away iff they are the same number of stops from Adelaide. But we are focussed here on the partial order induced by the actual railway network.) A strict partial order would be '$\underset{1}{____}$ is further along the line than $\underset{2}{____}$', which excludes ties.

A mathematical example of a partial order is the SUBSET relation $\subseteq$. Consider some set X which contains just a and b as members. The subsets of a given set X are those sets such that everything in them is also in X. So X is itself a subset of X; so is the set just containing a, and the set just containing b. Finally, the empty set (with no members) obviously is a subset of any set (everything in it is – trivially! – in every other set). That gives the structure of subsets depicted in Figure 21.14. The corresponding strict partial order $\subset$ results from removing any loops from a set to itself in that diagram. This is

7 Every station is no further along than itself; pick any two distinct stations, such as Woodville and Kilkenny: if Kilkenny is no further along than Woodville, then Woodville is further along than Kilkenny; and if we pick three stations, such as Oaklands, Brighton and Seacliff, then since Oaklands is no further than Brighton, and Bright is no further than Seacliff, it follows that Oaklands is no further than Seacliff.

obviously a partial order, since $\{a\}$ is neither a subset of $\{b\}$ or vice versa, though is is a subset of the original set $\{a, b\}$, and the empty set $\emptyset$ is a subset of both.

Key Ideas in §21

- An interpretation of Quantifier is given by a domain and a temporary assignment of extensions to some of the nonlogical vocabulary of Quantifier i.e., the names and predicates of the language.
- Any set of actual or merely possible things can serve as a domain.
- The extension of a name is a thing from some specified domain.
- The extension of an n-place predicate is a collection of ordered sequences of length n of things from the domain. The identity predicate has a privileged extension; it is always the collection of pairs from the domain consisting of things and themselves.
- The extension of a zero-place predicate is a truth value; these correspond to atomic sentences of Sentential.
- Given a domain, it is often possible to represent the interpretation of a predicate by means of an Euler diagram or a directed graph on that domain.
- Directed graphs provide a convenient way to represent the structural features of binary relations.

Practice exercises

A. For each of the following collections of individuals, properties and relations, construct an interpretation that includes any appropriate extensions they determine, and an appropriate domain. Use your personal judgment if needed, and comment on any difficulties.

1. The relation '$\underline{\quad}_1$ is just as rich as $\underline{\quad}_2$', Bill Gates, Elon Musk, and Warren Buffet;
2. The relation '$\underline{\quad}_1$ is exactly divisible by $\underline{\quad}_2$', and the property '$\underline{\quad}$ is even';
3. The property '$\underline{\quad}_1$ is a great novel', the relation '$\underline{\quad}_1$ is harder to understand than $\underline{\quad}_2$', *Moby Dick*, *Finnegan's Wake*, *Bleak House*, *A Wrinkle in Time*, *The Left Hand of Darkness*.
4. The relation '$\underline{\quad}_1$ is part of $\underline{\quad}_2$', your left leg, your lower body, you.

B. Since Quantifier is an extensional language, if let the Quantifier name 's' denote Superman, and the name 'c' denote Clark Kent, then '$s = c$' will be true. But it will be

true for the same reason that '$s = s$' is true: because ⟨Clark Kent, Clark Kent⟩ is in the extension of '='. Can you use this observation as the basis for any argument that English is not an extensional language?

C. Using some of the methods introduced in this section (Euler diagrams and directed graphs), give a graphical representation of the following interpretations. You will need to rely on your general knowledge to prepare these diagrams.

1.
domain: people
M: $\underbrace{____}_{1}$ is a musician;
W: $\underbrace{____}_{1}$ is a woman;
E: $\underbrace{____}_{1}$ is an economist.

2.
domain: cards in a standard deck
S: $\underbrace{____}_{1}$ is a seven;
F: $\underbrace{____}_{1}$ is a face card;
R: $\underbrace{____}_{1}$ is red;
j: the Jack of Hearts.

3.
domain: Australian states
L: $\underbrace{____}_{1}$ is larger in population than $\underbrace{____}_{2}$;
V: $\underbrace{____}_{1}$'s name ends in a vowel;
a: South Australia;
q: Queensland.

D. These questions concern the material from the optional section on binary relations (§21.10).

1. Show that any total relation on a domain is reflexive on that domain.
2. Show that a transitive relation is asymmetric if and only if it is irreflexive.
3. What is wrong with the following argument that reflexivity is a consequence of symmetry and transitivity?

   > If ⟨x,y⟩ is in $\mathfrak{R}$, then ⟨y,x⟩ is in $\mathfrak{R}$ since we assume $\mathfrak{R}$ is symmetric. If both ⟨x,y⟩ is in $\mathfrak{R}$ and ⟨y,x⟩ is in $\mathfrak{R}$, then since $\mathfrak{R}$ is transitive, ⟨x,x⟩ is in $\mathfrak{R}$ – so $\mathfrak{R}$ is reflexive.

4. A relation $\mathfrak{R}$ is SERIAL on a domain D iff for each x in D, there exists some y such that ⟨x,y⟩ is in $\mathfrak{R}$. Show that a serial, symmetric and transitive relation is reflexive.

22

Truth in Quantifier

We now know what interpretations are. Since, among other things, they tell us which predicates are true of which objects – and pairs, etc., of objects –, they provide us with an account of the truth of atomic sentences. But we must show how to extend that to an account of what it is for any Quantifier sentence to be true or false in an interpretation.

We know from §20 that there are three kinds of sentence in Quantifier:

- atomic sentences (i.e., atomic formulae of Quantifier which have no free variables);
- sentences whose main connective is a sentential connective; and
- sentences whose main connective is a quantifier.

We need to explain truth for all three kinds of sentence.

I shall offer a completely general explanation in this section. However, to try to keep the explanation comprehensible, I shall at several points use the following interpretation:

domain:	all people born before 2000CE
a:	Aristotle
b:	George W Bush
W:	$\underbrace{____}_{1}$ is wise
R:	$\underbrace{____}_{1}$ was born before $\underbrace{____}_{2}$

This will be my *go-to example* in what follows.

22.1 Atomic Sentences

The truth of atomic sentences should be fairly straightforward. The sentence 'Wa' should be true just in case 'W' is true of whatever is named by 'a'. Given our go-to

interpretation, this is true iff 'is wise' is true of whatever is named by 'Aristotle', i.e., iff Aristotle is wise. In fact (in the actual world) Aristotle is wise. So the sentence is true. Equally, 'Wb' is false on our go-to interpretation, because George W Bush is not wise.

Likewise, on this interpretation, 'Rab' is true iff the object named by 'a' was born before the object named by 'b'. Well, Aristotle was born before Bush. So 'Rab' is true. Equally, 'Raa' is false: Aristotle was not born before Aristotle. We can summarise these intuitive ideas more generally:

> › When $\mathcal{F}$ is a one-place predicate, and a is a name, then $\mathcal{F}a$ is true in an interpretation iff:
> the object assigned as the extension of a is among the objects assigned to the extension of $\mathcal{F}$.
>
> › When $\mathcal{R}$ is an n-place predicate and $a_1, a_2, ..., a_n$ are names (not necessarily all different from each other), $\mathcal{R}a_1a_2...a_n$ is true in an interpretation iff:
> the n-tuple of objects assigned as the extensions of $a_1, a_2, ..., a_n$ in that interpretation, $\langle a_1, ..., a_n\rangle$, is among the n-tuples comprising the extension assigned to $\mathcal{R}$ in that interpretation.

Two other kinds of atomic sentences exist: zero-place predicates, and identity sentences.

› A zero-place predicate is true in an interpretation iff it is assigned the extension true in that interpretation.

Here in fact, we are using Sentential valuations as constituents of Quantifier interpretations. (A valuation is what you get when you are only intepreting zero-place predicates; we will want it to turn out that the part of Quantifier which deals just with zero-place predicates and their truth-functional combinations is the familiar language Sentential, so those interpretations will have to behave like the valuations we are familiar with.)

› Identity sentences (two names flanking the identity predicate) are also easy to handle. Where a and b are any names, $a = b$ is true in an interpretation iff: a and b have the same extension (are assigned the very same object) in that interpretation.[1]

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

[1] Of course this is just the result of applying the conditions above for atomic sentences with two-place predicates to the special constant extension assigned to '=': $a = b$ is true iff the pair of the extensions of a and b is in the extension of identity, which can only happen if the extensions are the same.

22.2 Sentential Connectives

In Sentential, the truth value of a sentence in a valuation depends only on its main connective, and the truth values of its constituent sentences (§8.3). The truth value of a complex sentence 'percolates up' from those of its grammatically simpler constituents.

Things are exactly the same when it comes to Quantifier sentences that are built up from simpler ones using the truth-functional connectives that were familiar from Sentential. (We introduced such sentences in §20.) The rules governing these truth-functional connectives are *exactly* the same as they were when we considered Sentential. Here they are:

Where $\mathcal{A}$ and $\mathcal{B}$ are any sentences of Quantifier,

- $\mathcal{A} \wedge \mathcal{B}$ is true in an interpretation iff:
  both $\mathcal{A}$ is true and $\mathcal{B}$ is true in that interpretation
- $\mathcal{A} \vee \mathcal{B}$ is true in an interpretation iff:
  either $\mathcal{A}$ is true or $\mathcal{B}$ is true in that interpretation
- $\neg\mathcal{A}$ is true in an interpretation iff:
  $\mathcal{A}$ 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 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

This presents the very same information as the schematic 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 go-to interpretation:

- '$(a = a \wedge Wa)$' is true;
- '$(Rab \wedge Wb)$' is false because, although 'Rab' is true, 'Wb' is false;
- '$(a = b \vee Wa)$' is true;
- '$\neg a = b$' is true;
- '$(Wa \wedge \neg(a = b \wedge Rab))$' is true, because '$Wa$' is true and '$a = b$' is false.

Make sure you understand these examples.

22.3 When the Main Connective is a Quantifier

The exciting innovation in Quantifier, 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. The general principle of compositionality faces a challenge when it

comes to sentences whose main connective is a quantifier. We cannot say that the truth value of '$\exists x Fx$' depends on its syntax and the truth value of 'Fx', because we have no guidance in assigning a truth value to a formula with a free variable (though see §22.6).

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

Unfortunately, this naïve first thought is not general enough. For example, we want to be able to say that '$\forall x \exists y \mathcal{L}xy$' is true just in case '$\exists y \mathcal{L}xy$' is true of everything in the domain. And this is problematic, since our interpretation does not directly specify what '$\exists y \mathcal{L}xy$' is to be true of. Instead, whether or not this is true of something should follow just from the interpretation of $\mathcal{L}$, the domain, and the meanings of the quantifiers.

So here is a second naïve thought. We might try to say that '$\forall x \exists y \mathcal{L}xy$' is to be true in an interpretation iff $\exists y \mathcal{L}ay$ is true for *every* name a that we have included in our interpretation. And similarly, we might try to say that $\exists y \mathcal{L}ay$ is true just in case $\mathcal{L}a\mathscr{b}$ is true for *some* name $\mathscr{b}$ that we have included in our interpretation. (This kind of approach is known as SUBSTITUTIONAL QUANTIFICATION – our own approach below in §22.4 will make use of substitution, but in a more sophisticated way.)

Unfortunately, this is not right either. To see this, observe that in our go-to interpretation, we have only given interpretations for *two* names, 'a' and 'b'. But the domain – all people born before the year 2000CE – contains many more than two people. I have no intention of trying to name *all* of them! In most interpretations, things in the domain go unnamed; we can't understand quantifiers as ranging over only actually named things without missing out some things in such interpretations.[2]

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 previously uninterpreted name to our interpretation. I shall offer a few examples of how this might work, centring on our go-to interpretation, and I shall then present the formal definition.

› In our go-to interpretation, '$\exists x Rbx$' should be true. After all, in the domain, there is certainly someone who was born after Bush. Lady Gaga is one of those

[2] Can we solve this issue by the brute force proposal to simply assign everything in the domain a name? That is not possible, because there are not enough names. Remember (§20) that names in Quantifier consist of the English letters $a, \ldots, r$ with numerical subscripts if needed. So every name in Quantifier consists of a letter and a finite numeral. It turns out you can arrange all these names in a single, infinitely long list (basically, represent each name by a code number, and order the names by the size of the code number). The items in a list like that can be enumerated. But Cantor famously showed that some things are too many to be enumerated; any single list which attempts to include all of them would inevitably miss some. (One example: while you can enumerate all the finite sequences of itemse from a finite alphabet, you cannot enumerate all the infinite sequences of such items.) So some collections are too many for them all to have names, because we'd run out of names before labelling them all. So substitutional quantification can't handle all examples of quantification.

Cantor's result, and the 'diagonal argument' he used to establish it, is discussed in ch. 5 of Tim Button (2021) *Set Theory: An Open Introduction*, st.openlogicproject.org/settheory-screen.pdf.

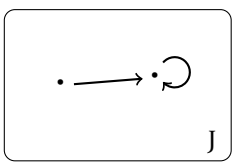
people. Indeed, if we were to extend our go-to interpretation – temporarily, mind – by adding the name 'c' to refer to Lady Gaga, then 'Rbc' would be true on this extended interpretation. And this, surely, should suffice to make '$\exists xRbx$' true on the original go-to interpretation.

- In our go-to interpretation, '$\exists x(Wx \wedge Rxa)$' should also be true. After all, in the domain, there is certainly someone who was both wise and born before Aristotle. Socrates is one such person. Indeed, if we were to extend our go-to interpretation by letting a previously uninterpreted name, 'c', denote Socrates, then '$Wc \wedge Rca$' would be true on this extended interpretation. Again, this should surely suffice to make '$\exists x(Wx \wedge Rxa)$' true on the original go-to interpretation.

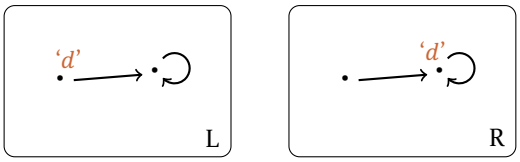
- In our go-to interpretation, '$\forall x\exists yRxy$' should be false. After all, consider the last person born in the year 1999. I don't know who that was, but if we were to extend our go-to interpretation by letting a previously uninterpreted name, 'd', denote that person, then we would not be able to find anyone else in the domain to denote with some further previously uninterpreted name, perhaps 'e', in such a way that 'Rde' would be true. Indeed, no matter *whom* we named with 'e', 'Rde' would be false. And this observation is surely sufficient to make '$\exists yRdy$' *false* in our extended interpretation. And this is sufficient to make '$\forall x\exists yRxy$' false on the original interpretation.

Look at the multiplying extensions; the quantifiers involved are mirrored in the interpretations we are asked to consider. '$\exists xFx$' is true if there *exists* an extended interpretation where some named thing is F, while '$\forall xFx$' is true if *every* extended interpretation makes the newly named thing F. In effect, we handle quantification over possibly unnamed objects in a domain by quantifying over potential interpretations that assign names to those objects while keeping everything else the same.

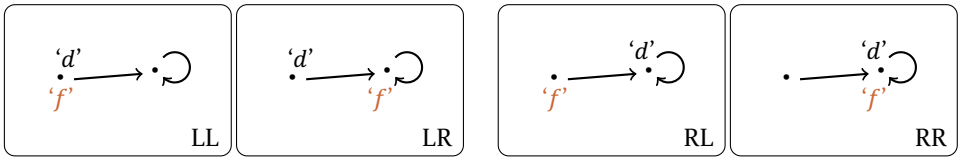
Some readers might prefer a more visual aid. If the domain is large, we need to consider *lots* of extended intepretations, because we need to consider assigning each of the items in the domain a new name. So I will consider a very simple interpretation J, with two objects and some binary relation between them, pictured here:



Note there are no labels on these nodes. No names are assigned to objects in this interpretation. The arrows give the extension of 'Q' in this interpretation J. Suppose we want to consider whether '$\exists x\forall yQyx$' is true in J. Then we want to know whether there is some way of assigning a new name 'd' to entities in this domain to make '$\forall yQyd$' come out true. So there are two interpretations to consider, L and R, because there are two things in the domain to which 'd' might be attached:



The original sentence is true in J iff '$\forall y Qyd$' is true in one of these extended interpretations. But in turn that is the case iff 'Qfd' is true in each interpretation extending the already extended interpretation by adding yet another new name 'f'. So now there are four interpretations to consider: two extensions of L, and two extensions of R:



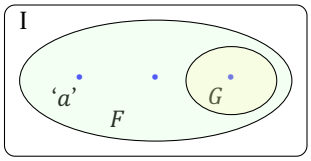
We now have all the interpretations we'll need. Let's go through it step by step:

1. '$\exists x \forall y Qyx$' is true in J iff
2. There exists *some* extended interpretation that assigns something to a newly chosen name 'd' such that '$\forall y Qyd$' is true in it, i.e., iff
3. Either '$\forall y Qyd$' is true in L or '$\forall y Qyd$' is true in R; ie., iff either
   a) 'Qfd' is true in every interpretation extending L, i.e., iff
      i. 'Qfd' is true in LL (no); and
      ii. 'Qfd' is true in LR (no).

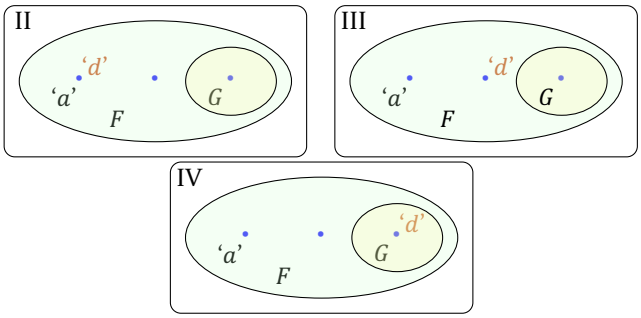
      OR
   b) 'Qfd' is true in every interpretation extending R, i.e., iff
      i. 'Qfd' is true in RL (yes); and
      ii. 'Qfd' is true in RR (yes).
4. Since 'Qfd' is true in both RL and RR, then
5. '$\forall y Qyd$' is true in R, so that
6. '$\exists x \forall y Qyx$' is true in J.

All of these interpretations that spawn from our original interpretation share the same domain, and the same extension for 'Q'. They differ only in that the extended interpretations add new labels to the items in the domain, interpreting the previously unused names.

Let's try another visual example. Consider this interpretation I, with a domain of three things, an interpreted name 'a' and two interpreted predicates 'F' and 'G'.



There are three extended interpretations, II–IV, corresponding to the three ways of adding some unused name 'd':



We can see that '$\forall x Fx$' is true in I, because in each of II–IV, 'd' labels something in the 'F'-region. We can see that '$\forall x Gx$' is false in I, because in II and III, 'd' labels something not in the 'G'-region. '$\exists x Gx$' is true in I, because interpretation IV attaches 'd' to something in the 'G'-region.

22.4 Formal Truth Conditions for Quantified Sentences

If you have understood the three examples, and the visual aids, you've understood everything that 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, x is a variable, and c is a name. We shall write '$\mathcal{A}|_{c\frown x}$' to represent the formula that results from replacing or SUBSTITUTING every free occurrence of x in $\mathcal{A}$ by c. So if we began with the formula 'Fxy', the metalanguage expression "'Fxy'$|_{c\frown x}$" denotes the formula 'Fcy'. If we began with 'Fyy', then 'Fyy'$|_{c\frown x}$ just is the original formula 'Fyy', since there are no instances of 'x' in 'Fyy' to be replaced. If we began with '$Fx \vee \forall x Gx$', then '$Fx \vee \forall x Gx$'$|_{c\frown x}$ is '$Fc \vee \forall x Gx$', since neither occurrence of 'x' in '$\forall x Gx$' is free.

Suppose we begin with a quantified formula, $\forall x\mathcal{A}$ or $\exists x\mathcal{A}$. If we strip off the quantifier, and pick any name c, then $\mathcal{A}|_{c\frown x}$ is known as a SUBSTITUTION INSTANCE of the original quantified formulae, and c may be 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', because '$\exists x(Ryx \leftrightarrow Fx)'|_{e\frown y}$ turns out to be '$\exists x(Rex \leftrightarrow Fx)$'.

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

> › $\forall x\mathcal{A}$ is true in an interpretation iff:
> $\mathcal{A}|_{c\frown x}$ is true in *every* interpretation that extends the original interpretation by assigning an object to some previously uninterpreted name c not appearing in $\mathcal{A}$ (without changing the original interpretation in any other way).
>
> › $\exists x\mathcal{A}$ is true in an interpretation iff:
> $\mathcal{A}|_{c\frown x}$ is true in *some* interpretation that extends the original interpretation by assigning an object to some previously uninterpreted name c not appearing in $\mathcal{A}$ (without changing the original interpretation in any other way).

That is: we pick a previously uninterpreted name that doesn't appear in $\mathcal{A}$.[3] We uniformly replace any *free* occurrences of the variable x in $\mathcal{A}$ by our previously uninterpreted name, which creates a substitution instance of '$\forall x\mathcal{A}$' and '$\exists x\mathcal{A}$'. Then if this substitution instance is true on every (respectively, some) way of adding an interpretation of the previously uninterpreted name to our existing interpretation, then '$\forall x\mathcal{A}$' (respectively, '$\exists x\mathcal{A}$') is true on that existing interpretation.

To be clear: all this is doing is formalising (very pedantically) the intuitive idea expressed above. The result is a bit ugly, and the final definition might look a bit opaque. Hopefully, though, the *spirit* of the idea is clear.

The trickiest part of all of this is keeping things straight when you have nested quantifiers, particularly quantifiers of different types. As above, when we considered '$\exists y\forall xQyx$', we needed first to consider interpretations that assigned some new name 'd', and then, for each of those interpretations, we generate a parasitic family of new interpretations assigning some other new name 'e'. If we apply our truth conditions, we can summarise the basic cases of two nested quantifiers as follows:

3 There will always be such a previously uninterpreted name: any given sentence of Quantifier only contains finitely many names, but Quantifier has a potentially infinite stock of names to draw from.

- $\forall x \forall y \mathcal{F}xy$: This will be true in interpretation $\mathfrak{I}$ iff for some new names 'c' and 'd', '$\mathcal{F}xy$'$|_{c \frown x}|_{d \frown y}$ – i.e., '$\mathcal{F}cd$' – is true in every interpretation just like $\mathfrak{I}$ except it also assigns extensions to the new names.
- $\exists x \exists y \mathcal{F}xy$ is true in $\mathfrak{I}$ iff for some new names 'c' and 'd', '$\mathcal{F}cd$' is true in some interpretation just like $\mathfrak{I}$ except it also assigns extensions to the new names.
- $\forall x \exists y \mathcal{F}xy$ is true in $\mathfrak{I}$ iff for some new names 'c' and 'd', for each interpretation $\mathfrak{I}'$ (otherwise like $\mathfrak{I}$) assigning an extension to 'c' there exists some interpretation $\mathfrak{I}''$ assigning an extension to 'd' (and otherwise just like $\mathfrak{I}'$) which makes '$\mathcal{F}cd$' true.
- $\exists x \forall y \mathcal{F}xy$ is true in $\mathfrak{I}$ iff for some new names 'c' and 'd', there exists some interpretation $\mathfrak{I}'$ (otherwise like $\mathfrak{I}$) assigning an extension to 'c' which makes '$\mathcal{F}cd$' true in each interpretation $\mathfrak{I}''$ assigning an extension to 'd' and otherwise just like $\mathfrak{I}'$.

Note that these rules are listed here for your convenience; they can be derived directly from the stipulation above, so you don't need to learn them separately.

22.5 Pitfalls of Alternative Approaches

Our rule for evaluating a universal quantifier sentence says $\forall x \mathcal{A}$ is true in an interpretation J iff $\mathcal{A}|_{c \frown x}$ is true in every interpretation just like J except that it assigns an extension to a previously unused name c. You might be thinking, *this all seems unnecessarily complicated. Can't we do things more simply?* Let's look at some simpler alternatives; we'll see that they go awry.

Why Restrict to New Names? Here is the first proposed simplification: do away with the requirement for the name in the extended interpretation to be new. That would give us this proposal:

- $\forall x \mathcal{A}$ is true in an interpretation J iff $\mathcal{A}|_{c \frown x}$ is true in any interpretation extending J which assigns an extension to 'c'.

This only differs from the correct proposal in cases where the name 'c' is already used. Suppose we consider '$\forall x c = x$'. This is false in any interpretation with two or more things – they can't both be c. But '$c = x$'$|_{c \frown x}$ is just '$c = c$'. And this is true in *every* interpretation which assigns anything to be c at all. Since '$c = c$' is true in every extended interpretation, this alternative rule *wrongly* predicts '$\forall x c = x$' is true in the original intepretation. The problem arises of course because the name we are substituting for the universally quantified variable interacts with the existing occurences of the name. The moral: always use a new previously uninterpreted name.

Why not substitute x for c? The truth conditions above start with a quantified sentence, drop the quantifier, and substitute a name for the associated variable. Why do things that way? Couldn't we start by considering the truth values of some sentence with a name across many interpretations, and then swap the name for a variable and add a quantifier? Here is a second proposed alternative:

> $\forall x\mathcal{A}|_{x\curvearrowright c}$ is true in an interpretation I iff in every interpretation just like I except that it interprets a new name c, $\mathcal{A}$ is true.

The problem here is that $\mathcal{A}$ is actually not well-defined. Consider this case, a sentence with a degenerate initial quantifier: '$\forall x\exists xRxx$'. The official truth conditions say: this sentence is true iff '$\exists xRxx$' is true in every extended interpretation that assigns something to a new name. Since '$\exists xRxx$' has no names, it will be true on all of these extended intepretations iff it is true in the original interpretation. So the initial $\forall x$ quantifier is redundant.

What happens on the alternative approach? It turns out there are several candidates for $\mathcal{A}$:

1. $\forall x\exists xRxx = \forall x\exists xRxx|_{x\curvearrowright c}$, so $\mathcal{A} = \exists xRxx$;
2. $\forall x\exists xRxx = \forall x\exists xRxc|_{x\curvearrowright c}$, so $\mathcal{A} = \exists xRxc$;
3. $\forall x\exists xRxx = \forall x\exists xRcc|_{x\curvearrowright c}$, so $\mathcal{A} = \exists xRcc$.

The problem is, these different candidates for $\mathcal{A}$ don't all give the same results in a given interpretation. ('$\exists xRxx$' certainly doesn't always have the same truth value as '$\exists xRxc$'.) This means that this doesn't actually provide a way of assigning a truth value to a quantified sentence. We need our definition of truth to determine an unambiguous answer, and this proposal doesn't manage to meet that requirement.

22.6 Satisfaction

The discussion of truth in §§22.1–22.4 only ever involved assigning truth values to sentences. When confronted with formulae involving free variables, we altered them by substituting a previously uninterpreted name for the variable, converting it into a sentence, and temporarily extending the interpretation to cover that previously uninterpreted name. This is a significant departure from the definition of truth for Sentential sentences in §8.3. There, we showed how the truth value of a complex Sentential sentence in a valuation depended on the truth value of its constituents in that same valuation. By contrast, on the approach just outlined, the truth value of '$\exists xFx$' in an interpretation depends on the truth value of some other sentence 'Fc', which is not a constituent of '$\exists xFx$', in some different (although related) interpretation!

There is another way to proceed, which allows arbitrary formulae of Quantifier, even those with free variables, to be assigned (temporarily) a truth value. This approach can let the truth value of '$\exists xFx$' in an interpretation depend on the temporary truth value of 'Fx' in that same interpretation. This is conceptually neater (with no multiplying

interpretations) than the approach just introduced, and I present it briefly here as an alternative to the substitutional approach of the preceding sections.

› *This section should be regarded as optional, and only for the dedicated student of logic.*

The inspiration for the approach comes, once again, from thinking of variables in Quantifier as behaving like pronouns in English. In §15.5 we gave a gloss of 'Someone is angry' as 'Some person is such that: they are angry'. Concentrate on 'they are angry'. This sentence featuring a bare pronoun doesn't express any specific claim intrinsically. But we can, temporarily, fix a referent for the pronoun 'they' – temporarily elevating it to the status of a name. If we do so, the sentence can be evaluated. We can introduce a referent by pointing: 'They [points to someone] are angry'. Or we can fix a referent by simply assigning one: 'Consider that person over there. They are angry'. If we can find someone or other to temporarily be the referent of the pronoun 'they', then it will be true that there is someone such that they are angry. If no matter who we fix as the referent, they are angry, then it will be true that everyone is such that they are angry.

This is, in a nutshell, the idea we will use to handle quantification in Quantifier. Let us introduce some terminology:

A variable assignment over an interpretation is an assignment of exactly one object from the domain of that interpretation to each variable that we care to consider.

If we have an interpretation, and a variable assignment, then we can evaluate every formula of Quantifier – not only sentences. Of course, the evaluation of the open formulae will be very fragile, since even given a single background interpretation, a formula like 'Fx' might be true relative to one variable assignment and false relative to another.

Let us start, as before, by giving rules for evaluating atomic formulae of Quantifier, given an interpretation and a variable assignment.

Where $\mathcal{R}$ is any n-place predicate ($n \geq 0$), and $t_1, ..., t_n$ are any terms – variables or names, then:

› $\mathcal{R}t_1...t_n$ is true on a variable assignment over an interpretation iff:
$\mathcal{R}$ is true of the objects assigned to the terms $t_1, ..., t_n$ by that variable assignment and that intepretation, considered in that order.

› $t_1 = t_2$ is true on a variable assignment over an interpretation iff:
t_1 and t_2 are assigned the very same object by that variable assignment in that interpretation.

These are very similar clauses to those we saw for atomic sentences in §22.4. Indeed, when we are considering atomic sentences of Quantifier, the mention of a variable assignment is redundant, since no atomic sentence contains a variable. (If an atomic formula contains a variable, the variable would be free and the formula thus not a sentence.)

The recursion clauses that extend truth for atomic formulae to truth for arbitrary formulae are these (I omit the clauses for $\vee$, $\rightarrow$, and $\leftrightarrow$, which you can easily fill in yourself, following the model in §22.2):

Where $\mathcal{A}$ and $\mathcal{B}$ are formulae of Quantifier, and x is a variable:

- $\neg\mathcal{A}$ is true on a variable assignment over an interpretation iff:
  $\mathcal{A}$ is false on that variable assignment over that interpretation;
- $\mathcal{A} \wedge \mathcal{B}$ is true on a variable assignment over an interpretation iff:
  both $\mathcal{A}$ is true on that variable assignment over that interpretation and $\mathcal{B}$ is true on that variable assignment over that interpretation;
- ...
- $\forall x\mathcal{A}$ is is true on a variable assignment over an interpretation iff:
  $\mathcal{A}$ is true on *every* variable assignment differing from the original one at most in what it assigns to x over that interpretation;
- $\exists x\mathcal{A}$ is is true on a variable assignment over an interpretation iff:
  $\mathcal{A}$ is true on *some* variable assignment differing from the original one at most in what it assigns to x over that interpretation.

The last two clauses are where this approach is strongest. Rather than considering some substitition instance of $\forall x\mathcal{A}$, we simply consider the direct constituent $\mathcal{A}$. Rather than considering all variations on the original interpretation which include some previously uninterpreted name, we simply consider all ways of varying what is assigned to x by the original variable assignment, but keeping everything else unchanged. If you see the rationale for the clauses offered in §22.4, you can see why the clauses just offered are appropriate.

We now have the idea of truth on a variable assignment over an intepretation. But what we want – if this alternative approach is to yield the same end result – is truth in an interpretation. Notice that, given an interpretation, varying the variable assignment can change the truth value of a formula with free variables. But it cannot change the truth value of a formula which is a sentence, so if a sentence is true on one variable assignment over an interpretation, it is true on every variable assignment over that interpretation. So we can reintroduce the notion of truth in an intepretation, like so:

> $\mathcal{A}$ is true in an interpretation iff:
> $\mathcal{A}$ is a sentence of Quantifier, and $\mathcal{A}$ is true on any (or every) variable assignment over that interpretation.

Here's how this works in practice. Suppose we want to figure out whether the sentence '$\forall x \exists y Lxy$' is true on an interpretation which associates the two-place predicate 'L' with the relation '$\underset{1}{\underline{\quad\quad}}$ is no heavier than $\underset{2}{\underline{\quad\quad}}$', and has as its domain the planets in our solar system. We might reason as follows:

> *For each way of picking planets to be the values of 'x' and 'y', either we pick two different planets, and one is lighter than the other (no two planets have the same mass); or we pick the same planet, and 'they' are identical in mass. In either case, we can always find something no heavier than anything we pick. So for any variable assignment to 'x' over this interpretation, we can then assign something to 'y' so as to make 'Lxy' true on that joint assignment. Hence no matter what we assign to 'x', '$\exists y Lxy$' is true on that assignment. But since that is true no matter what we assign to 'x', '$\forall x \exists y Lxy$' is true on every variable assignment over this interpretation. But that latter is a sentence, so is true in this interpretation.*

Let us say that a sequence of objects $\langle a_1, ..., a_n \rangle$ SATISFIES a formula $\mathcal{A}$ in which the variables $x_1, ..., x_n$ occur freely iff there is a variable assignment over an intepretation whose domain includes each a_i, and which assigns each a_i to the variable x_i, and on which $\mathcal{A}$ is true over that interpretation. What we have expressed in terms of variable assignments could have been expressed, a little more awkwardly, using the notion of some objects satisfying a formula. Indeed, this is how Alfred Tarski, the inventor of this approach to truth in Quantifier, first introduced the idea.[4]

[4] A translation of his original 1933 paper is Alfred Tarski (1983) 'The Concept of Truth in Formalized Languages' in his *Logic, Semantics, Metamathematics*, Indianapolis: Hackett, pp. 152–278. It is quite technical in places.

Key Ideas in §22

- An atomic sentence of Quantifier is true in an interpretation iff the extensions of the names occurring in it, taken in the appropriate order, fall in the extension of the predicate they accompany.
- A compound sentence of Quantifier has truth conditions relative to an interpretation that are the same as those for Sentential, if the main connective is truth-functional.
- A compound sentence of Quantifier whose main connective is a quantifier '$\forall x$' (resp., '$\exists x$') is true in an interpretation if on every (resp., some) interpretation extending the first by assigning an extension to some previously uninterpreted name 'c', the result of replacing every occurence of the variable 'x' bound by the quantifier with c is true.

Practice exercises

A. Consider the following interpretation:

domain: The domain comprises only Corwin and Benedict
A: is to be true of both Corwin and Benedict
B: is to be true of Benedict only
N: is to be true of no one
c: is to refer to Corwin

Determine whether each of the following sentences is true or false in that interpretation:

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

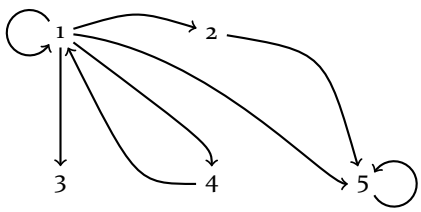
B. Consider the following interpretation:

domain: The domain comprises only Lemmy, Courtney and Eddy
G: is to be true of Lemmy, Courtney and Eddy.
H: is to be true of and only of Courtney
M: is to be true of and only of Lemmy and Eddy
c: is to refer to Courtney
e: is to refer to Eddy

Determine whether each of the following sentences is true or false in that interpretation:

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

C. Following the diagram conventions introduced at the end of §21, consider the following interpretation:



Determine whether each of the following sentences is true or false in that interpretation:

1. $\exists x Rxx$
2. $\forall x Rxx$
3. $\exists x \forall y Rxy$
4. $\exists x \forall y Ryx$
5. $\forall x \forall y \forall z((Rxy \wedge Ryz) \rightarrow Rxz)$
6. $\forall x \forall y \forall z((Rxy \wedge Rxz) \rightarrow Ryz)$
7. $\exists x \forall y \neg Rxy$
8. $\forall x(\exists y Rxy \rightarrow \exists y Ryx)$
9. $\exists x \exists y(\neg x = y \wedge Rxy \wedge Ryx)$
10. $\exists x \forall y(Rxy \leftrightarrow x = y)$
11. $\exists x \forall y(Ryx \leftrightarrow x = y)$
12. $\exists x \exists y(\neg x = y \wedge Rxy \wedge \forall z(Rzx \leftrightarrow y = z))$

D. Why, when we are trying to figure out whether '$\forall x Rxa$' is true in an interpretation, do we need to consider whether 'Rca' is true in some expanded interpretation

with a *new* name 'c'. Why can't we make do with substituting a name we've already interpreted?

E. Explain why on page 208 we did not give the truth conditions for the existential quantifer like this:

> $\exists x\mathcal{A}|_{x \curvearrowright c}$ is true in an interpretation I iff in *some* interpretation just like I except that might assign something different to c, $\mathcal{A}$ is true.

23

Semantic Concepts

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

So:

- An Quantifier sentence $\mathcal{A}$ is a LOGICAL TRUTH iff $\mathcal{A}$ is true in every interpretation; this is written $\vDash \mathcal{A}$.
- $\mathcal{A}$ is a LOGICAL FALSEHOOD iff $\mathcal{A}$ is false in every interpretation; i.e., $\vDash \neg\mathcal{A}$.
- Two Quantifier sentences $\mathcal{A}$ and $\mathcal{B}$ are LOGICALLY EQUIVALENT 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 Quantifier sentences $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are JOINTLY CONSISTENT iff there is some interpretation in which all of the sentences are true. They are JOINTLY INCONSISTENT iff there is no such interpretation.

Some examples might be helpful to illustrate these ideas.

Consider the sentences '$\forall x(Fx \lor Gx)$' and '$\neg(Fa \to Ga)$'. I offer an interpretation showing them consistent:

Domain: musical artists
F: $\underset{1}{\underline{\quad\quad}}$ plays an instrument
G: $\underset{1}{\underline{\quad\quad}}$ sings
a: Keith Moon

'Fa' is true in this interpretation, while 'Ga' is false. So the conditional '$(Fa \rightarrow Ga)$' is false, and so '$\neg(Fa \rightarrow Ga)$' is true. Suppose we add a previously unused name b to this interpretation: it will either denote a musician or a singer (I am including producers as playing a musical instrument). So $(Fb \vee Gb)$ will be true in each such extended interpretation. $(Fb \vee Gb)$ is $(Fx \vee Gx)|_{b \frown x}$, so by our semantic clauses, $\forall x(Fx \vee Gx)$ is true in the original interpretation. Since there is an interpretation making each of these sentences true, they are consistent.

Now look at '$\exists y(Py \wedge \neg Py)$'. If this is true in an intepretation, assigning some extension to 'P', then some interpretation with the same extension assigned to 'P' and some new name 'b' makes '$(Pb \wedge \neg Pb)$' true. But that can be the case only if the extension of 'b' is included in the extension of 'P', and also isn't included in that extension – impossible. So whatever extension we assign to 'P', '$\exists y(Py \wedge \neg Py)$' is false. So that sentence is false on every interpretation, and is a logical falsehood. Its negation is therefore a logical truth: '$\neg\exists y(Py \wedge \neg Py)$'. This is entirely general: the negation of a logical truth (falsehood) is a logical falsehood (truth).

Look now at '$\forall x \neg Gx$' and '$\neg\exists x Gx$'. Any interpretation which makes the first true must be such that any extended interpretation with a new name 'b' makes '$\neg Gb$' true, i.e., any extended interpretation will make 'Gb' false. Hence, there is no extended interpretation making 'Gb' true; so '$\exists x Gb$' is false in the original interpretation, hence '$\neg\exists x Gx$' is true. Each step in this line of argument can also be run in reverse; so '$\forall x \neg Gx$' and '$\neg\exists x Gx$' are true in exactly the same interpretations. They are logically equivalent.

Consider finally '$\forall x Gx$' and '$Ga \rightarrow \exists y(Py \wedge \neg Py)$'. These are inconsistent. Any interpretation making '$Ga \rightarrow \exists y(Py \wedge \neg Py))$' true must make '$Ga$' false – if it didn't, it would have to make its consequent true, but that's a logical falsehood. But could '$\forall x Gx$' be true in such an interpretation? No – that would require every extended interpretation to make 'Gb' true, even on the extended interpretation where the new name 'b' is assigned the same referent as the existing name 'a', which is not in the extension of 'G' in this interpretation. So no interpretation can make both of these sentences true.

Given this, any interpretation which makes '$Ga \rightarrow \exists y(Py \wedge \neg Py)$' true must make '$\forall x Gx$' false, and must therefore make '$\neg\forall x Gx$' true. So there is no interpretation on which '$Ga \rightarrow \exists y(Py \wedge \neg Py)$' is true and in which '$\neg\forall x Gx$' is false. This is a familiar notion from Sentential: entailment. We will use the symbol '$\vDash$' for entailment in Quantifier, much as we did for Sentential. This introduces an ambiguity, but a harmless one, since every valid argument in Sentential remains a valid argument in Quantifier.

> $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ together ENTAIL $\mathcal{C}$ (written $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{C}$) iff
>
> - every interpretation on which all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are true is also one in which $\mathcal{C}$ is true; or, equivalently,
> - 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.

These definitions establish a close connection between consistency and entailment, as before: $\mathcal{A}_1, \ldots, \mathcal{A}_n \vDash \mathcal{C}$ iff $\mathcal{A}_1, \ldots, \mathcal{A}_n$, and $\neg\mathcal{C}$ are jointly inconsistent. Since joint

inconsistency is a property of some sentences collectively, it doesn't actually matter which of them is regarded as 'the' conclusion. The truth of all but one of them will force the remaining one to be false, whichever one is chosen. So the joint inconsistency of '$Ga \rightarrow \exists y(Py \wedge \neg Py)$' and '$\forall x Gx$' establishes also that $\forall x Gx \vDash \neg(Ga \rightarrow \exists y(Py \wedge \neg Py))$.

Also as before, there is a close connection between validity and entailment:

> $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_n \therefore \mathcal{C}$ is VALID in Quantifier 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 Quantifier otherwise.

Every possible situation determines some interpretation (assuming it is not possible for there to be nothing at all), since every possible situation will have a domain and its properties and relations will determine extensions on that domain. So whenever there is an entailment, there is no interpretation (and hence no possible situation) making the premise true and the conclusion false. So the argument must be conclusive.

Quantifier entailments are conclusive in virtue of form, because the consideration of many alternative interpretations of the nonlogical expressions involved means that no argument can be conclusive because of some special 'connection in meaning' between such expressions in the language. Entailment in Quantifier reflects only the logical structure of the sentences involved, where this now includes the new logical resources of Quantifier over Sentential.

Still, just as with Sentential, there are arguments that are valid in English that aren't symbolised as a valid argument in Quantifier. Consider

It is raining
So: It will be that it was raining.

This is conclusive: there is no possible situation in which it is raining now, but in the future it turns out that it never was raining. But the best we can apparently do in quantifier is to symbolise 'it is raining' by 'P', and 'it will be that it was raining' by 'Q', and $P \therefore Q$ is not valid in Quantifier. If this argument is valid, it will depend on the logic of expressions that are treated as non-logical expressions in Quantifier– here, the tenses 'is', 'was' and 'will be'. We will look at this example again briefly in §38.2.

23.1 Some Subtleties about Truth and Interpretation

Our interpretations allow us a precise characterisation of formal truth for Quantifier. In §2, we introduced the idea of the structure of a sentence, and the pragmatic approach that logicians take to that notion. In Quantifier, the logical constants are those of Sentential plus the quantifiers and the identity predicate. But every other item of the language is up for reinterpretation, and our interpretations allow us to reinterpret these expressions in an arbitrary fashion.

In Sentential, we were able to characterise a logical truth as a sentence which is actually true on every reinterpretation of nonlogical expressions (in Sentential, the nonlogical

expressions are just are the atomic sentences). And we characterised a formally valid argument as one in which each possible reinterpretation (i.e., valuation) of the premises which makes them actually true, is also one which makes the conclusion actually true.

But this account of formal validity needs refining when it comes to Quantifier. Consider the Quantifier sentence which says that there are at least three things:

$$\exists x \exists y \exists z (x \neq y \wedge x \neq z \wedge y \neq z).$$

This sentence contains no nonlogical expressions. Hence the sentence has a constant meaning, and is true on each reinterpretation just in case it is actually true. It is actually true – there are actually at least three things. Hence our Sentential-inspired account of logical truth would suggest that this sentence is a logical truth. But it seems very strange to think that it is a *logical* truth that there are at least three things. Isn't it possible that there had been fewer? And shouldn't logic allow for that possibility?

The keen-eyed among you will have noticed that this sentence is not, in fact, a logical truth of Quantifier. The reason is that in defining truth for sentences of Quantifier, we considered not just reinterpretations of the nonlogical expressions, but also we allowed the domain of our interpretation to freely depart from actuality (§21.6). So we are, in effect, allowing our interpretations to vary the meanings of the nonlogical vocabulary *and* to vary the possible situations at which our sentences are to be evaluated. So we can consider a possible situation in which there are just two things, and if that situation provides the domain of an interpretation, there is no way of extending that interpretation to three names 'a', 'b', and 'c' such that '$a \neq b \wedge a \neq c \wedge b \neq c$' is true; hence '$\exists x \exists y \exists z (x \neq y \wedge x \neq z \wedge y \neq z)$' isn't true on the original interpretation.

One sentence of this class is nevertheless a logical truth: the claim that something exists, '$\exists x x = x$'. Since it is a constraint on Quantifier domains that they cannot be empty (§15.6), there is for any interpretation a way of adding a previously uninterpreted name 'c' which denotes an object in the domain, and of course '$c = c$' is true in that extended interpretation, since each name must have a referent in any interpretation, and the identity predicate is always interepreted as representing pairs of objects in the domain and themselves.

This appeal to possible situations is very suggestive. Every possible situation seems to have a domain of things which exist in that possibility, and the properties and relations that are instantiated in that possibility determine extensions drawn from that domain. So you might wonder: should we think of interpretations as just being 'possible worlds'? Should we, that is, think that a sentence $\mathcal{A}$ is consistent (i.e., there is an interpretation on which it is true) iff $\mathcal{A}$ is true in some possible situation?

We should not. In Quantifier it is not even clear how to do this, because a sentence like '$\exists x (Fx \wedge Gx)$' doesn't even seem to mean anything without a prior intepretation. If we assign a meaning to 'F' and 'G', we can then see if there are possible situations making the sentence true. The problem then will be that there are logically consistent sentences that are true in no possible world. For example, suppose we interpret 'F' to be '_____ is a cat' and 'G' to be '_____ is a mineral'. Obviously, given these asssignments of meaning, the sentence can be rendered in English as 'there is a cat which is a mineral',

and that is impossible: I would assume that it is of the essence of a cat that it be a living animal, not an inanimate mineral. But the sentence is perfectly consistent despite being impossible, precisely consistency involves reinterpreting the predicates so that they can possibly hold together. There is a significant difference between possibility and consistency:

- Possibility is about the meaning of an already interpreted sentence: holding that meaning fixed, is there a possible situation which it describes?
- Consistency is about logical structure, according to some language: holding that structure fixed, is there a way to assign a meaning to the non-structural expressions to make the sentence come out true?

This distinction is not always noted. Philosophers are prone to talking of 'logical possibility' where they generally mean to talk about consistency. Possibility, as I understand it, is a property of intepreted sentences, or propositions. Logical consistency is a property of interpretable sentences, relative to some language in which they are assigned a meaning. In this sense, the logical 'languages' Sentential and Quantifier are more like proto-languages, because only a select few of the expressions of the language are antecedently meaningful.[1]

Key Ideas in §23

- Definitions of entailment, consistency, logical falsehood and logical equivalence can be given for Quantifier that are very close parallels to those for Sentential, though given in terms of intepretations not valuations.
- The notion of a logical truth in Quantifier is slightly different than the notion of a logical truth in Sentential, since we allow interpretations that vary not just in what they assign as extensions to the nonlogical vocabulary, but also variation in the domain over which quantifiers range. Nothing like the latter feature occurs in Sentential valuations, but it is crucial for ensuring that contingent counting sentences, without names or nonlogical predicates, aren't misclassified as logical truths. Consistency must be distinguished from possibility.

Practice exercises

A. The following entailments are correct in Quantifier. In each case, explain why.

[1] I owe my understanding of this point to the distinction between 'representational' and 'interpretational' semantics drawn by John Etchemendy (1999) *The Concept of Logical Consequence*, CSLI Publications, chs. 2–5.

1. $\forall x Qx, \forall x(Qx \to Rx) \vDash \exists x Rx$;
2. $\forall x \forall y\big(Rxy \to \forall z(Ryz \to x \neq z)\big) \vDash (Raa \to \neg Raa)$.
3. $\exists x(Px \wedge Qx) \vDash Pa$;
4. $\exists x Rxx \vDash \exists x \exists y Rxy$;
5. $\exists x \forall y Rxy \vDash \forall y \exists x Rxy$;
6. $\forall x \forall y \forall z((x \neq y \wedge x \neq z) \to y = z) \vDash \neg \exists x \exists y \exists z(x \neq y \wedge (x \neq z \wedge y \neq z))$.

B. Show that, for any formula $\mathcal{A}$ with at most x free, the following two sentences are logically equivalent: $\exists x \mathcal{A}$ and $\neg \forall x \neg \mathcal{A}$.

C. Show that

1. $\mathcal{A} \vDash \mathcal{B}$ iff '$\mathcal{A} \to \mathcal{B}$' is a logical truth;
2. $\mathcal{A}$ is logically equivalent to $\mathcal{B}$ iff '$\mathcal{A} \leftrightarrow \mathcal{B}$' is a logical truth.

24

Demonstrating Consistency and Invalidity

24.1 Logical Truths and Logical Falsehoods

Suppose we want to show that '$\exists x Axx \rightarrow Bd$' is *not* a logical truth. This requires showing that the sentence is not true in every interpretation; i.e., that it is false in some interpretation. If we can provide just one interpretation in which the sentence is false, then we will have shown that the sentence is not a logical truth.

In order for '$\exists x Axx \rightarrow Bd$' 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 offer:

A: $\underset{1}{\underline{\quad\quad}}$ is in the same place as $\underset{2}{\underline{\quad\quad}}$

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: $\underset{1}{\underline{\quad\quad}}$ is in Germany

Now we have an interpretation where '$\exists xAxx$' is true, but where 'Bd' is false. So there is an interpretation where '$\exists xAxx \rightarrow Bd$' is false. So '$\exists xAxx \rightarrow Bd$' is not a logical truth.

We can just as easily show that '$\exists xAxx \rightarrow Bd$' is not a logical falsehood. We need only specify an interpretation in which '$\exists xAxx \rightarrow Bd$' is true; i.e., an interpretation in which either '$\exists xAxx$' is false or 'Bd' is true. Here is one:

domain: Paris
d: Paris
A: $\underbrace{____}_{1}$ is in the same place as $\underbrace{____}_{2}$
B: $\underbrace{____}_{1}$ is in France

This shows that there is an interpretation where '$\exists xAxx \rightarrow Bd$' is true. So '$\exists xAxx \rightarrow Bd$' is not a logical falsehood.

24.2 Logical Equivalence

Suppose we want to show that '$\forall xSx$' and '$\exists xSx$' are not logically equivalent. We need to construct an interpretation in which the two sentences have different truth values; we want one of them to be true and the other to be false. We start by specifying a domain. Again, we make the domain small so that we can specify extensions easily. In this case, we 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, Sarah Vaughan

We can make '$\exists xSx$' true by including something in the extension of 'S', and we can make '$\forall xSx$' false by leaving something out of the extension of 'S'. For concreteness we shall offer:

S: $\underbrace{____}_{1}$ plays saxophone

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

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

24.3 Validity, Entailment and Consistency

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

Consider the following argument in Quantifier:

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

To show that this is invalid, we must make the premise true and the conclusion false. The conclusion is a conditional, so to make it false, the antecedent must be true and the consequent must be false. Clearly, our domain must contain two objects. Let's try:

domain: Karl Marx, Ludwig von Mises
G: $\underbrace{____}_{1}$ 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 '$\exists xGx$' is true in this interpretation. Hence '$\exists xGx \to Ga$' is false, as required.

But does this interpretation make the premise true? Yes it does! For '$\exists x(Gx \to Ga)$' to be true, '$Gc \to Ga$' must be true in some extended interpretation that is almost exactly like the interpretation just given, except that it also interprets some previously uninterpreted name 'c'. Let's extend our original interpretation by letting 'c' denote Karl Marx – the same thing as 'a' denotes in the original interpretation. Since 'a' and 'c' denote the same thing in the extended interpretation, obviously '$Gc \to Ga$' will be true. So '$\exists x(Gx \to Ga)$' is true in the original interpretation. So the premise is true, and the conclusion is false, in this original interpretation. The argument is therefore invalid.

In passing, note that we have also shown that '$\exists x(Gx \to Ga)$' does *not* entail '$\exists xGx \to Ga$'. And equally, we have shown that the sentences '$\exists x(Gx \to Ga)$' and '$\neg(\exists xGx \to Ga)$' are jointly consistent.

Let's consider a second example. Consider:

$$\forall x\exists yLxy \therefore \exists y\forall xLxy$$

Again, I want to show that this is invalid. To do this, we must make the premises true and the conclusion false. Here is a suggestion:

domain: People with a living biological sibling
L: $\underbrace{____}_{1}$ shares a parent with $\underbrace{____}_{2}$

The premise is clearly true on this interpretation. Anyone in the domain has a living sibling. That sibling will also, then, be in the domain, because one cannot be someone's sibling without also having them as a sibling. So for everyone in the domain, there will be at least one other person in the domain who is their sibling, and thus has a parent in common with them. 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 shares a parent with everyone in the domain, and there is no such person. So the argument is invalid. We observe immediately that the sentences '$\forall x \exists y Lxy$' and '$\neg \exists y \forall x Lxy$' are jointly consistent and that '$\forall x \exists y Lxy$' does not entail '$\exists y \forall x Lxy$'.

For my third example, I shall mix things up a bit. In §21, I described how we can present some interpretations using diagrams. For example:

$$\begin{tikzcd}
1 \arrow[loop left] \arrow[r] \arrow[dr] & 2 \arrow[d] \\
 & 3
\end{tikzcd}$$

Using the conventions employed in §21, the domain of this interpretation is the first three positive whole numbers, and 'R' is true of x and y just in case there is an arrow from x to y in our diagram. Here are some sentences that the interpretation makes true:

- '$\forall x \exists y Ryx$'
- '$\exists x \forall y Rxy$' witness 1
- '$\exists x \forall y (Ryx \leftrightarrow x = y)$' witness 1
- '$\exists x \exists y \exists z (\neg y = z \wedge Rxy \wedge Rzx)$' witness 2
- '$\exists x \forall y \neg Rxy$' witness 3
- '$\exists x (\exists y Ryx \wedge \neg \exists y Rxy)$' witness 3

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

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

and many more besides.

To show that $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \therefore \mathcal{C}$ is invalid, 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 $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ do not entail $\mathcal{C}$.
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 – that some sentence is a logical truth, say – this is sometimes called providing a COUNTERMODEL.

Key Ideas in §24

- Testing for consistency and invalidity involves displaying just a single appropriate interpretation.
- It can be a subtle matter to figure out what such an interpretation might be like.
- An interpretation which shows an argument invalid is a counterexample to that argument.

Practice exercises

A. Show that each of the following is neither a logical truth nor a logical falsehood:

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. $\exists x(Gx \to \forall y My)$
7. $\exists x(x = h \land x = i)$

B. For each of the following, say whether the sentence is a logical truth, a logical falsehood, or neither:

1. $\exists x \exists y x = y$;
2. $(\forall x(Px \land Qx) \leftrightarrow \forall x Px \land \forall x Qx)$;
3. $\exists x(Px \land Qx) \to \neg(\exists x Px \land \exists x Qx)$;
4. $\forall x \forall y(x \neq y \to (Fx \leftrightarrow \neg Fy))$.

C. Show that the following pairs of sentences are not logically equivalent.

1. Ja, Ka
2. $\exists x Jx, Jm$
3. $\forall x Rxx, \exists x Rxx$
4. $\exists x Px \to Qc, \exists x(Px \to Qc)$
5. $\forall x(Px \to \neg Qx), \exists x(Px \wedge \neg Qx)$
6. $\exists x(Px \wedge Qx), \exists x(Px \to Qx)$
7. $\forall x(Px \to Qx), \forall x(Px \wedge Qx)$
8. $\forall x \exists y Rxy, \exists x \forall y Rxy$
9. $\forall x \exists y Rxy, \forall x \exists y Ryx$

D. Show that the following sentences are jointly consistent:

1. $Ma, \neg Na, Pa, \neg Qa$
2. $Lee, Leg, \neg Lge, \neg Lgg$
3. $\neg(Ma \wedge \exists x Ax), Ma \vee Fa, \forall x(Fx \to Ax)$
4. $Ma \vee Mb, Ma \to \forall x \neg Mx$
5. $\forall y Gy, \forall x(Gx \to Hx), \exists y \neg Iy$
6. $\exists x(Bx \vee Ax), \forall x \neg Cx, \forall x\big((Ax \wedge Bx) \to Cx\big)$
7. $\exists x Xx, \exists x Yx, \forall x(Xx \leftrightarrow \neg Yx)$
8. $\forall x(Px \vee Qx), \exists x \neg(Qx \wedge Px)$
9. $\exists z(Nz \wedge Ozz), \forall x \forall y(Oxy \to Oyx)$
10. $\neg \exists x \forall y Rxy, \forall x \exists y Rxy$
11. $\neg Raa, \forall x(x = a \vee Rxa)$
12. $\forall x \forall y \forall z(x = y \vee y = z \vee x = z), \exists x \exists y\ \neg x = y$
13. $\exists x \exists y(Zx \wedge Zy \wedge x = y), \neg Zd, d = e$

E. Show each of the following non-entailments by providing an appropriate interpretation on which the premises are true and the conclusion false:

1. $Pa \nvDash \exists x(Px \wedge Qx)$;
2. $\forall y(Py \to \exists x Ryx) \nvDash \forall x(Px \to \exists y Ryy)$;
3. $\forall x Rxx \nvDash \forall x Rax$.

F. Show that the following arguments are invalid:

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

25

Reasoning about All Interpretations: Demonstrating Inconsistency and Validity

25.1 Logical Truths and Logical Falsehoods

We can show that a sentence is *not* a logical truth just by providing one carefully specified interpretation: an interpretation in which the sentence is false. To show that something is a logical truth, on the other hand, it would not be enough to construct ten, one hundred, or even a thousand interpretations in which the sentence is true. A sentence is only a logical truth if it is true in *every* interpretation, and there are infinitely many interpretations. We need to reason about all of them, and we cannot do this by dealing with them one by one!

Sometimes, we can reason about all interpretations fairly easily. For example, we can offer a relatively simple argument that '$Raa \leftrightarrow Raa$' is a logical truth:

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

This argument is valid, of course, and its conclusion is true. However, it is not an argument in Quantifier. Rather, it is an argument in English *about* Quantifier: it is an argument in the metalanguage.

Note another feature of the argument. Since the sentence in question contained no quantifiers, we did not need to think about how to interpret 'a' and 'R'; the point was just that, however we interpreted them, 'Raa' would have some truth value or other. (We could ultimately have given the same argument concerning Sentential sentences.)

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

> Consider some arbitrary interpretation. Consider some arbitrary member of the model's domain, which, for convenience, we shall call *obbie*, and suppose we extend our original interpretation by adding a previously uninterpreted name, 'c', to name *obbie*. Then either 'Rcc' will be true or it will be false. If 'Rcc' is true, then '$Rcc \leftrightarrow Rcc$' is true. If 'Rcc' is false, then '$Rcc \leftrightarrow Rcc$' will be true. So either way, '$Rcc \leftrightarrow Rcc$' is true. Since there was nothing special about *obbie* – we might have chosen any object – we see that no matter how we extend our original interpretation by allowing 'c' to name some new object, '$Rcc \leftrightarrow Rcc$' will be true in the new interpretation. So '$\forall x(Rxx \leftrightarrow Rxx)$' was true in the original interpretation. But we chose our interpretation arbitrarily. So '$\forall x(Rxx \leftrightarrow Rxx)$' is true in every interpretation. It is therefore a logical truth.

This is quite longwinded, but, as things stand, there is no alternative. In order to show that a sentence is a logical truth, we must reason about *all* interpretations.

But sometimes we can draw a conclusion about some way all interpretations must be, by considering a hypothetical interpretation which isn't that way, and showing that hypothesis leads to trouble. Here is an example.

> Suppose there were an interpretation which makes '$a \neq b \rightarrow \neg\exists x(x = a \wedge x = b)$' false. Then while '$a \neq b$' is true on that interpretation, '$\neg\exists x(x = a \wedge x = b)$' must be false. So '$\exists x(x = a \wedge x = b)$' must be true; and hence it has some true substitition instance with some previously uninterpreted name 'c' as the instantiating name: '$c = a \wedge c = b$' is true. But then it must also be that '$a = b$' is true, which contradicts our original supposition that '$a \neq b$' is true on this intepretation. So there can be no such interpretation, and hence the sentence cannot be false on any interpretation and must be a logical truth.

This is an example of *REDUCTIO* reasoning: we assume something, and derive some absurdity or contradiction; we then conclude that our assumption is to be rejected. It is often easier to show that all Fs have a property by deriving an absurdity from the assumption that some F lacks the property, than it is to show it 'directly' by considering each F in turn. This is a powerful technique, used widely in mathematics and elsewhere.

25.2 Other Cases

Similar points hold of other cases too. Thus, we must reason about all interpretations if we want to show:

- that a sentence is a logical falsehood; for this requires that it is false in *every* interpretation.
- that two sentences are logically equivalent; for this requires that they have the same truth value in *every* interpretation.
- that some sentences are jointly inconsistent; for this requires that there is no interpretation in which all of those sentences are true together; i.e., that, in *every* interpretation, at least one of those sentences is false.
- that an argument is valid; for this requires that the conclusion is true in *every* interpretation where the premises are true.
- that some sentences entail another sentence.

The problem is that, with the tools available to you so far, reasoning about all interpretations is a serious challenge! Let's take just one more example. Here is an argument which is obviously valid:

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

After all, if everything is both H and J, then everything is H. But we can only show that the argument is valid 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 previously uninterpreted 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 '$\forall xHx$' 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 '$\forall xHx$' is true. The argument is valid!

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

Reductio reasoning is particularly useful in demonstrating the validity of valid arguments. In that case we will typically make the hypothetical supposition that there is some interpretation which makes the premises true and the conclusion false, and then derive an absurdity from that supposition. From that we can conclude there is no such intepretation, and hence that any interpretation which makes the premises true must also be one which makes the conclusion true.

The following table summarises whether a single (counter-)interpretation suffices, or whether we must reason about all interpretations (whether directly considering them all, or indirectly making use of *reductio*).

	Yes	No
logical truth?	all interpretations	one counter-interpretation
logical falsehood?	all interpretations	one counter-interpretation
logically equivalent?	all interpretations	one counter-interpretation
consistent?	one interpretation	consider all interpretations
valid?	all interpretations	one counter-interpretation
entailment?	all interpretations	one counter-interpretation

This might usefully be compared with the table at the end of §13.1. The key difference resides in the fact that Sentential concerns truth tables, whereas Quantifier concerns interpretations. This difference is deeply important, since each truth-table only ever has finitely many lines, so that a complete truth table is a relatively tractable object. By contrast, there are infinitely many interpretations for any given sentence(s), so that reasoning about all interpretations can be a deeply tricky business.

Key Ideas in §25

- To demonstrate that something is a logical truth or a logical falsehood, or that an argument is an entailment, involves reasoning about all intepretations.
- Reasoning about all intepretations of a given sentence is intrinsically more difficult than reasoning about all valuations, because there are only finitely many valuations to consider for any Sentential sentence, but infinitely many intepretations for an Quantifier sentence.
- Some indirect strategies, like *reductio*, can be very useful in overcoming this obstacle.

Practice exercises

A. Show that each of the following is either a logical truth or a logical falsehood:

1. $Da \land \neg Da$;
2. $Pm \land \neg \exists x Px$;
3. $\forall x \forall y (x \neq y \leftrightarrow y \neq x)$;
4. $\forall x (Fx \lor \exists y \neg Fy))$;
5. $\exists x ((x = h \land x = i) \land (Fh \leftrightarrow \neg Fi))$.

B. Show that the following pairs of sentences are logically equivalent.

1. $\exists x (Jx \land x = m)$, Jm;
2. $\forall x \forall y (x = y \land Rxy)$, $\exists x (Rxx \land \forall y (x = y)$;
3. $(\exists x Px \land Qc)$, $\exists x (Px \land Qc)$;
4. $\forall x \forall y Rxy$, $\forall x \forall y Rxy$;
5. $\forall x (Px \rightarrow \exists y Rxy)$, $\neg \exists x (\forall y \neg Rxy \land Px)$.

C. Show that the following sentences are jointly inconsistent:

1. $Ma \lor Mb, (Ma \to \forall x \neg Mx), \neg Mb$;
2. $\exists x(Bx \lor Ax), \forall x \neg Cx, \forall x\big((Ax \lor Bx) \to Cx\big)$;
3. $\forall x \forall y(Rxy \to \neg Ryx), \forall x Rxx$.
4. $\forall x \forall y x = y, \exists x Px, \neg \forall x Px$.

D. Show that the following arguments are valid:

1. $Ma \therefore (Qa \lor \neg Qa)$;
2. $\neg(Mb \land \exists x Ax), (Mb \land Fb) \therefore \neg \forall x(Fx \to Ax)$;
3. $\forall y Gy, \exists y \neg Hy \therefore \neg \forall x(Gx \to Hx)$;
4. $\exists x Kx, \forall x(Kx \leftrightarrow \neg Lx) \therefore \exists x \neg Lx$;
5. $\forall x Qx, \forall x(Qx \to Rx) \therefore \exists x Rx$;
6. $\exists z(Nz \land Ozz) \therefore \neg \forall x \forall y(Oxy \to \exists z(\neg Nz \land (z = x \lor z = y)))$;
7. $\exists x \exists y(Zx \land Zy \land x \neq y), \neg Zd \therefore \exists y(Zy \land y \neq d)$;
8. $\forall x \forall y\,(Rxy \to \forall z(Ryz \to x \neq z)) \therefore Raa \to \neg Raa$.

Chapter 6

Natural Deduction for Sentential

26

Proof and Reasoning

26.1 Arguments and Reasoning Revisited

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

In the case of Sentential, this led us to develop truth tables. Each row of a complete truth table corresponds to a valuation. So, when faced with a Sentential 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 plod through the truth table.

But truth tables do not necessarily give us much *insight*. Consider two arguments in Sentential:

$$P \lor Q, \neg P \therefore Q$$
$$P \leftrightarrow Q, P \therefore Q.$$

Clearly, these are valid arguments. You can confirm that they are valid by constructing four-row truth tables. But we might say that two arguments, proceeding from different premises with different logical connectives involved, must make use of different principles of implication. What follows from a disjunction is not at all the same as what follows from a biconditional, and it might be nice to keep track of these differences. While a truth table can show *that* an argument is valid, it doesn't really explain why the argument is valid. To explain why $P \lor Q, \neg P \therefore Q$ is valid, we have to say something about how disjunction and negation work and interact.

Certainly human reasoning treats disjunctions very differently from biconditionals. While logic is not really about human reasoning, which is more properly a subject matter for psychology, nevertheless we can formally study the different forms of reasoning involved in arguing from sentences with different structures, by asking: *what would it be acceptable to conclude from a premise with a certain formal structure, supposing one cannot give up one's committment to that premise?*

With truth tables, we only really care about the truth values assigned to whole sentences, since that is what ultimately determines whether there is an entailment.

With natural deduction systems, we are thinking more directly about reasoning *from* premises *to* conclusions. We see what the premises ultimately commit us to, and see whether those consequences are sufficient to establish the conclusion. To do this, we manipulate sentences in accordance with rules that we regard as acceptable. Indeed, with just a small 'starter pack' of rules of implication, just characterising what a sentence implies, or is implied by, in virtue of its main connective, we might hope to capture all valid arguments.

This is a very different way of thinking about arguments.

Natural deduction systems promise to give us a better insight – or at least, a different insight – into how arguments work. Consider this pair of arguments:

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

From a truth-table perspective, these arguments are indistinguishable: their premises are true in the same valuations, and so are their conclusions. Yet they are distinct from an implicational point of view, since one will involve consideration of the consequences of disjunctions and negations, and the other consideration of the consequences of the conditional. So the natural deduction perspective on arguments promises to give us a more fine-grained analysis of how valid arguments function.

26.2 Efficiency in Natural Deduction

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

$$A_1 \to C_1 \therefore (A_1 \land A_2) \to (C_1 \lor C_2).$$

To test this argument for validity, you might use a 16-row truth table. If you do it correctly, then you will see that there is no row on which all the premises are true and on which the conclusion is false. So you will know that the argument is valid. (But, as just mentioned, there is a sense in which you will not know *why* the argument is valid.) But now consider:

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

This argument is also valid – as you can probably tell – but to test it requires a truth table with $2^{20} = 1048576$ rows. In principle, we can set a machine to grind through truth tables and report back when it is finished. In practice, complicated arguments in Sentential can become *intractable* if we use truth tables. But there is a very short natural deduction proof of this argument – just 6 rows. You can see it on page 260, though it won't make much sense if you skip the intervening pages.

When we get to Quantifier, though, the problem gets dramatically worse. There is nothing like the truth table test for Quantifier. To assess whether or not an argument is valid,

we have to reason about *all* interpretations. But there are infinitely many possible interpretations. We cannot even in principle set a machine to grind through infinitely many possible interpretations and report back when it is finished: it will *never* finish. We either need to come up with some more efficient way of reasoning about all interpretations, or we need to look for something different. Since we already have some motivation for considering the role in arguments of particular premises, rather than all the premises collectively as in the truth-table approach, we will opt for the 'something different' path – natural deduction.

26.3 Our System of Natural Deduction and its History

Rather than reasoning directly about all valuations (in the case of `Sentential`) or all interpretations (in the case of `Quantifier`), we shall try to select a few basic rules of implication. Some of these will govern the behaviour of the sentential connectives. Others will govern the behaviour of the quantifiers and identity. The resulting system of rules will give us a new way to think about the validity of arguments.

The modern development of natural deduction dates from simultaneous and unrelated papers by Gerhard Gentzen and Stanisław Jaśkowski in the 1930s.[1] However, the natural deduction system that we shall consider is based on slightly later work by Frederic Fitch.[2]

Natural deduction was so-called because the rules of implication it codifies were seen as reflecting 'natural' forms of human reasoning. It must be admitted that no one spontaneously and instinctively reasons in a way that conforms to the rules of natural deduction. But there is one place where these forms of inference are widespread – mathematical reasoning. And it will not surprise you to learn that these systems of inference were introduced initially to codify good practice in mathematical proofs. Don't worry, though: we won't expect that you are already a fluent mathematician. Though some of the rules might be a bit stilted and formal for everyday use, the rationale for each of them is transparent and can be easily understood even by those without extensive mathematical training.

One further thing about the rules we shall give is that they are extremely *simple*. At every stage in a proof, it is trivial to see which rules apply, how to apply them, and what the result of applying them is. While constructing proofs as a whole might take some thought, the individual steps are the kind of thing that can be undertaken by a completely automated process. The development of formal proofs in the early years of the twentieth century emphasised this feature, as a part of a general quest to remove the need for 'insight' or 'intuition' in mathematical reasoning. As the philosopher and

[1] Gerhard Gentzen (1934) 'Untersuchungen über das logische Schließen', translated as 'Investigations into Logical Deduction' in M. E. Szabo, ed. (1969) *The Collected Works of Gerhard Gentzen*, North-Holland. Stanisław Jaśkowski (1934) 'On the rules of suppositions in formal logic', reprinted in Storrs McCall, ed. (1967) *Polish logic 1920–39*, Oxford University Press.

[2] Frederic Fitch (1952) *Symbolic Logic: An introduction*, Ronald Press Company.

In the design of the present proof system, I drew on earlier versions of `forall`x, but also on the natural deduction systems of Jon Barwise and John Etchemendy (1992) *The Language of First-Order Logic*, CSLI; and of Volker Halbach (2010) *The Logic Manual*, Oxford University Press.

mathematician Alfred North Whitehead expressed his conception of the field, 'the ultimate goal of mathematics is to eliminate any need for intelligent thought'. You will see I hope that natural deduction does require some intelligent thought. But you will also see that, because the steps in a proof are trivial and demonstrably correct, that finding a formal proof for a claim is a royal road to mathematical knowledge.

Key Ideas in §26

- A formal proof system can be theoretically useful, as it might give insight into *why* an argument is valid by showing *how* the conclusion can be derived from the premises.
- It might also be practically useful, because it can be much faster to produce a single proof demonstrating an entailment than it would be to show that *all* valuations or interpretations making the premises true also make the conclusion true.
- A natural deduction proof system aims to use natural and obviously correct rules, which can contribute to the project of establishing the conclusions of proofs as certain knowledge.

27

The Idea of a Formal Proof: Reasoning from Assumptions

We will develop a NATURAL DEDUCTION system. The fundamental idea is that a formal proof involves reasoning from *assumptions* – seeing which sentences follows from which, in virtue of their main connectives. For each connective, there will be INTRODUCTION rules, that allow us to prove a sentence that has that connective as the main connective, and ELIMINATION rules, that allow us to prove something given a sentence that has that connective as the main connective.

Very abstractly, any FORMAL PROOF is a sequence of sentences, each accompanied by a 'commentary' on that sentence. There are lots of different systems for constructing formal proofs in Sentential, including axiomatic systems, semantic tableaux, Gentzen-style natural deduction, and sequent calculus. Different systems may be more efficient, or simpler to use, or better when one is reasoning about formal proofs rather than constructing them. The system we use here is best for some purposes – it is easy to understand that the rules are correct, it is fairly straightforward, and it has the nice feature that one can reuse one proof within the course of constructing another. All formal proof systems share two important features:

- The rules are unambiguous; and
- The rules apply at some stage of a proof only because of the syntactic structure of the sentences involved.. This makes them ideal for computers to use.

These two features make it easy to design a computer program that produces formal proofs, as long as it can parse and analyse expressions of Sentential. No consideration of meanings need be involved, which makes it quite unlike the earlier ways we had of analysing arguments in Sentential, such as truth-tables, which essentially involved understanding the truth-functions that characterise the meanings of the logical connectives of the language.

In a natural deduction system, some of those sentences may be initial ASSUMPTIONS, or suppositions that we are making 'for the sake of argument'. Other sentences will follow from earlier sentences in the formal proof by applying certain rules we will detail below. The commentary tells us whether the sentence is an assumption, or whether it follows from one of the rules from an earlier sentence or sentences, and specifies which sentence(s) it follows from, and which assumptions it relies on. The last line of the formal proof is the conclusion. Some of the initial assumptions may remain as assumptions or suppositions even at the end of the formal proof – these UNDISCHARGED ASSUMPTIONS will be the premises of an argument that the formal proof demonstrates to be valid and whose conclusion is the last line of the proof. Henceforth, I shall simply call these 'proofs', but you should be aware that there are *informal proofs* too.[1]

We will use a particular graphical representation of natural deduction proofs, one which makes use of 'nesting' of sentences to vividly represent which assumptions a particular sentence in a proof is relying on at any given stage, and uses a device of horizontal marks to distinguish assumptions from derived sentences. It will be easier to see how this works by considering some examples!

So consider:

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

We shall start a proof by writing an assumption:

1	$\neg(A \vee B)$

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

Note also that we have drawn a line underneath the premise. Everything written above the line is an *assumption*. Everything written below the line will either be something which follows from the assumptions we have already made, or it will be some new assumption.

The vertical line at the side marks the scope of this assumption: any subsequent sentence which is adjacent to a continuation of this vertical line is understood to lie in the RANGE of this assumption. We might call these vertical lines ASSUMPTION LINES. If some line in a proof is in the range of an assumption, we will say that it is an ACTIVE ASSUMPTION at that point in the proof. If we are going to write down some sentence that follows from an assumption, we will involve first extending the corresponding assumption line to indicate that it rests on this assumption, or at least is playing a role in the argument connected with this assumption. As we will see, not every sentence in the range of an assumption will intuitively *depend* on the assumption, but many will.

We are hoping to conclude that '$\neg A \wedge \neg B$'; so we are hoping ultimately to conclude our proof with

[1] Many of the arguments we offer in our metalanguage, quasi-mathematical arguments *about* our formal languages, are proofs. Sometimes people call formal proofs 'deductions' or 'derivations', to ensure that no one will confuse the metalanguage activity of proving things about our logical languages with the activity of constructing arguments within those languages. But it seems unlikely that anyone in this course will be confused on this point, since we are not offering very many proofs in the metalanguage in the first place!

1	$\neg(A \vee B)$
	$\vdots$
n	$\neg A \wedge \neg B$

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

Suppose we wanted to consider an argument with more than one premise:

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

If our argument has more than one premise, we can write more than one assumption down on successive lines, as long as the small horizontal line comes after all of the assumptions we are making at some stage. The argument has three premises, so we start by writing them all down, numbered, and drawing a line under them. We are hoping to conclude with $C \vee D$ at the end, so we put that down too, though we cannot yet fill in the middle of the proof:

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

Note that the single vertical line to the left indicates the range of all of the assumptions lying above the small horizontal line. We could, if we want, have introduced each of the premises as a separate assumption, with its own range:

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

This can sometimes be useful when we want to see which conclusions follow from some but not all of the premises. It looks a bit different to the earlier proof, but in fact represents the same dependencies: the conclusion is in the range of three assumptions, all of which remain in force throughout the proof.

What remains to do is to explain each of the rules that we can use along the way from premises to conclusion. The rules are divided into two families: those rules that involve

making or getting rid of further assumptions that are made 'for the sake of argument', and those that do not. The latter class of rules are simpler, so we will begin with those. After introducing the rules, I will return in §29.10 to the two incomplete proofs above, to see how they may be completed.

Key Ideas in §27

- Any formal proof is a sequence of sentences, each of which follows by some relatively simple rules from previous sentences or is licensed in some other way. The final sentence is the conclusion of the proof.
- The rules are unambiguous and apply only because of the syntactic structure of the sentences involved: no consideration of meanings need be involved. This makes them ideal for computers to use.
- A formal natural deduction proof is a graphical representation of reasoning from assumptions, in accordance with a strict set of rules for deriving further claims and managing which assumptions are active ('undischarged') at a given point in the proof.

Practice exercises

A. If the following a correctly formed proof in our natural deduction system?

$$
\begin{array}{l|l}
1 & A \\
2 & ((B \leftrightarrow) \vee A) \\
\hline
3 & \quad \begin{array}{|l} \neg A \\ \hline \end{array} \\
4 & \quad \quad \begin{array}{|l} D \rightarrow \neg D \\ \hline \end{array}
\end{array}
$$

B. Which of the following could, given the right rules, be turned into a proof corresponding to the argument

$$\neg(P \wedge Q), P \therefore \neg Q?$$

1.
$$
\begin{array}{l|l}
1 & \neg(P \wedge Q) \\
\hline
2 & \quad \begin{array}{|l} P \\ \hline \\ \vdots \\ \\ \neg Q \end{array} \\
n &
\end{array}
$$

2.

1	$\neg(P \wedge Q)$
2	$\quad P$
	$\quad \vdots$
n	$\neg Q$

3.

1	$\neg Q$
	$\vdots$
n	P
$n+1$	$\neg(P \wedge Q)$

28

Basic Rules for Sentential: Rules without Subproofs

28.1 Conjunction Introduction

Suppose I want to show that Ludwig is reactionary *and* libertarian. One obvious way to do this would be as follows: first I show that Ludwig is reactionary; then I show that Ludwig is libertarian; 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 to represent the argument in Sentential:

R: Ludwig is reactionary
L: Ludwig is libertarian

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

8	R	
	$\vdots$	
15	L	
16	$(R \wedge L)$	$\wedge$I 8, 15

Note that every line of our proof must either be an assumption, or must be justified by some rule. We add the commentary '$\wedge$I 8, 15' here to indicate that the line is obtained by the rule of conjunction introduction ($\wedge$I) applied to lines 8 and 15. Note the derived conjunction depends on the collective assumptions of the two conjuncts.

Since the order of conjuncts does not matter in a conjunction, I could equally well have obtained '$(L \wedge R)$' as '$(R \wedge L)$'. I can use the same rule with the commentary reversed, to reflect the reversed order of the conjuncts:

8	R	
	$\vdots$	
15	L	
16	$(L \wedge R)$	$\wedge$I 15, 8

More generally, here is our conjunction introduction rule: if we have obtained $\mathcal{A}$ and $\mathcal{B}$ by some stage in a proof under some shared assumptions – whether by proof or assumption – that justifies us in introducing their conjunction, which inherits those same assumptions. Abstractly, the rule looks like this:

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

To be clear, the statement of the rule is *schematic*. It is not itself a proof. '$\mathcal{A}$' and '$\mathcal{B}$' are not sentences of Sentential. Rather, they are symbols in the metalanguage, which we use when we want to talk about any sentence of Sentential (see §7). Similarly, 'm' and 'n' are not a numerals that will appear on any actual proof. Rather, they are symbols in the metalanguage, which we use when we want to talk about any line number of any proof. In an actual proof, the lines are numbered '1', '2', '3', and so forth. But when we define the rule, we use variables to 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, earlier than the line that results from the application of the rule. They can be separated from one another, and they can appear in any order. So m might be less than n, or greater than n. Indeed, m might even equal n, as in this proof:

1	P	
2	$P \wedge P$	$\wedge$I 1, 1

Note that the rule involves extending the vertical line to cover the newly introduced sentence. This is because what has been derived depends on the same assumptions as what it was derived from, and so it must also be in the range of those assumptions.

> All of the rules in this section justify a new claim which inherits all the assumptions of anything from which it has been derived by a natural deduction rule.

The two starting conjuncts needn't have the same assumptions, but the derived conjunction inherits their joint assumptions:

1	P	
2	$\quad Q$	
3	$\quad (P \land Q)$	$\land$I 1, 2

28.2 Conjunction Elimination

The above rule is called 'conjunction *introduction*' because it introduces a sentence with '$\land$' as its main connective into our proof, prior to which it may have been absent. Correspondingly, we also have a rule that *eliminates* a conjunction. Not that the earlier conjunction is somehow removed! It's just that we use a sentence whose main connective is a conjunction to justify further sentences in which that conjunction does not feature.

Suppose you have shown that Ludwig is both reactionary and libertarian. You are entitled to conclude that Ludwig is reactionary. Equally, you are entitled to conclude that Ludwig is libertarian. Putting these observations together, we obtain our CONJUNCTION ELIMINATION rules:

m	$(\mathcal{A} \land \mathcal{B})$		m	$(\mathcal{A} \land \mathcal{B})$	
	$\vdots$			$\vdots$	
	$\mathcal{A}$	$\land$E m		$\mathcal{B}$	$\land$E m

The point is simply that, when you have a conjunction on some line of a proof, you can obtain either of the conjuncts by $\land$E later on. There are two rules, because each conjunction justifies us in deriving either of its conjuncts. We could have called them $\land$E-LEFT and $\land$E-RIGHT, to distinguish them, but in the following we will mostly not distinguish them.[1]

One point might be worth emphasising: you can only apply this rule when conjunction is the main connective. So you cannot derive 'D' just from '$C \lor (D \land E)$'! Nor can you

[1] Why do we have two rules at all, rather than one rule that allows us to derive either conjunct? The answer is that we want our rules to have an unambiguous result when applied to some prior lines of the proof. This is important if, for example, we are implementing a computer system to produce formal proofs.

derive 'D' directly from '$C \wedge (D \wedge E)$', because it is not one of the conjuncts of the main connective of this sentence. You would have to first obtain '$(D \wedge E)$' by ∧E, and then obtain 'D' by a second application of that rule, as in this proof:

1	$C \wedge (D \wedge E)$	
2	$D \wedge E$	∧E 1
3	D	∧E 2

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

$$\begin{array}{ll} & \big((A \vee B) \rightarrow (C \vee D)\big) \wedge \big((E \vee F) \rightarrow (G \vee H)\big) \\ \therefore & \big((E \vee F) \rightarrow (G \vee H)\big) \wedge \big((A \vee B) \rightarrow (C \vee D)\big) \end{array}$$

Dealing with this argument using truth-tables would be a very tedious exercise, given that there are 8 sentence letters in the premise and we would thus require a $2^8 = 256$ line truth table! But we can deal with it swiftly using our natural deduction rules.

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

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

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

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

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

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

This is a very simple proof, but it shows how we can chain rules of proof together into longer proofs. Our formal proof requires just four lines, a far cry from the 256 lines that would have been required had we approached the argument using the techniques from chapter 3.

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

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

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

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

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

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

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

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

Recall that our official definition of sentences in Sentential only allowed conjunctions with two conjuncts. When we discussed semantics, we became a bit more relaxed, and allowed ourselves to drop inner parentheses in long conjunctions, since the order of the parentheses did not affect the truth table. The proof just given suggests that we could also drop inner parentheses in all of our proofs. However, this is not standard, and we shall not do this. Instead, we shall return to the more austere parenthetical

conventions. (Though we will allow ourselves to drop outermost parentheses most of the time, for legibility.)

Our conjunction rules correspond to intuitively correct patterns of implication. But they are also demonstrably good in another sense. Each of our rules can be vindicated by considering facts about entailment. Each of these schematic entailments is easily demonstrated:

- $\mathcal{A}, \mathcal{B} \vDash \mathcal{A} \land \mathcal{B}$;
- $\mathcal{A} \land \mathcal{B} \vDash \mathcal{A}$;
- $\mathcal{A} \land \mathcal{B} \vDash \mathcal{B}$.

For example, the first of these says that $\mathcal{A}$ and $\mathcal{B}$ separately suffice to entail the truth of their conjunction. This justifies the proof rule of conjunction introduction, since at a stage in the proof where we are assuming both $\mathcal{A}$ and $\mathcal{B}$ to be true, we are then permitted to conclude that $\mathcal{A} \land \mathcal{B}$ is true – just as conjunction introduction says we can.

It can be recognised, then, that our proof rules correspond to valid arguments in Sentential, and so our conjunction rules will never permit us to derive a false sentence from true sentences. There is no guarantee of course that the assumptions we make in our formal proofs are in fact true – only that if they were true, what we derive from them would also be true. So despite the fact that our proof rules are a *syntactic* procedure, that rely only on recognising the main connective of a sentence and applying an appropriate rule to introduce or eliminate it, each of our rules corresponds to an acceptable entailment.

28.3 Conditional Elimination

Consider the following argument:

> If Jane is smart then she is fast. Jane is smart. So Jane is fast.

This argument is certainly valid. If you have a conditional claim, that commits you to the consequent given the antecedent, and you also have the antecedent, then you have sufficient material to derive the consequent.

This suggests a straightforward conditional elimination rule (→E):

m	$(\mathcal{A} \to \mathcal{B})$	
	$\vdots$	
n	$\mathcal{A}$	
	$\vdots$	
	$\mathcal{B}$	$\to$E m, n

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

Here is an illustration of the rules we have so far in action, applied to this intuitively correct argument:

$$P, \big((P \to Q) \land (P \to R)\big) \therefore (R \land Q).$$

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

The correctness of our proof rule of conditional elimination is supported by the easily demonstrated validity of the corresponding entailment:

› $\mathcal{A}, \mathcal{A} \to \mathcal{B} \vDash \mathcal{B}$.

So applying this rule can never produce false conclusions if we began with true assumptions.

28.4 Biconditional Elimination

The BICONDITIONAL ELIMINATION rule ($\leftrightarrow$E) lets you do a much the same as the conditional rule. Unofficially, a biconditional is like two conditionals running in each

direction. So, thought of informally, our biconditional rules correspond to the left-to-right conditional elimination rule, the other of which corresponds to a right-to-left application of conditional elimination.

If we know that Alice is coming to the party iff Bob is, then if we knew that either of them was coming, we'd know that the other was coming. 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 have these two instances of the rule:

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

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

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

Here is an example of the biconditional rules in action, demonstrating the following argument:

$$P, (P \leftrightarrow Q), (Q \rightarrow R) \therefore R.$$

1	$(P \leftrightarrow Q)$	
2	$\quad P$	
3	$\quad\quad (Q \rightarrow R)$	
4	$\quad\quad Q$	$\leftrightarrow$E 1, 2
5	$\quad\quad R$	$\rightarrow$E 3, 4

Note the way that our conjunction and conditional elimination rules can be used to parallel the biconditional elimination rules:

1	$((P \rightarrow Q) \wedge (Q \rightarrow P))$	
2	Q	
3	$(Q \rightarrow P)$	$\wedge$E 1
4	P	$\rightarrow$E 3, 2

1	$(P \leftrightarrow Q)$	
2	Q	
3	P	$\leftrightarrow$E 1, 2

The correctness of our proof rules of biconditional elimination is supported by the easily demonstrated validity of the corresponding entailments:

- $\mathcal{A} \leftrightarrow \mathcal{B}, \mathcal{A} \vDash \mathcal{B}$;
- $\mathcal{A} \leftrightarrow \mathcal{B}, \mathcal{B} \vDash \mathcal{A}$.

28.5 Disjunction Introduction

Suppose Ludwig is reactionary. Then Ludwig is either reactionary or libertarian. After all, to say that Ludwig is either reactionary or libertarian is to say something weaker than to say that Ludwig is reactionary. ($\mathcal{A}$ is weaker than $\mathcal{B}$ if $\mathcal{A}$ follows from $\mathcal{B}$, but not *vice versa*.)

Let me emphasise this point. Suppose Ludwig is reactionary. It follows that Ludwig is *either* reactionary *or* a kumquat. Equally, it follows that *either* Ludwig is reactionary *or* that kumquats are the only fruit. Equally, it follows that *either* Ludwig is reactionary *or* that God is dead. Many of these things would be strange *inferences* to draw. Since the truth of the assumption does guarantee that the disjunction is true, there is nothing *logically* wrong with the implications. This can be so even if drawing these implications may violate all sorts of implicit conversational norms, or that inferring in accordance with logic in this manner would be more likely a sign of psychosis than rationality.

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

m	$\mathcal{A}$		m	$\mathcal{A}$	
	$\vdots$			$\vdots$	
	$(\mathcal{A} \vee \mathcal{B})$	$\vee$I m		$(\mathcal{B} \vee \mathcal{A})$	$\vee$I m

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

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

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

Here is an example, to show our rules in action:

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

This disjunction rule is supported by the following valid Sentential argument forms:

- $\mathcal{A} \vDash \mathcal{A} \lor \mathcal{B}$; and
- $\mathcal{B} \vDash \mathcal{A} \lor \mathcal{B}$.

28.6 Reiteration

The last natural deduction rule in this category is REITERATION (R). This just allows us to repeat an assumption or claim $\mathcal{A}$ we have already established, so long as the current stage of the proof remains in the range of any assumption which $\mathcal{A}$ is in the range of.

m	$\mathcal{A}$	
	$\vdots$	
	$\mathcal{A}$	R m

Such a rule is obviously legitimate; but one might well wonder how such a rule could ever be useful. Here is an example of it in action:

1	P	
2	$((P \land P) \to Q)$	
3	P	R 1
4	$(P \land P)$	$\land$I 1, 3
5	Q	$\to$E 2, 4

This rule is unnecessary at this point in the proof (we could have applied conjunction introduction and cited line 1 twice in our commentary), but it can be easier in practice to have two distinct lines to which to apply conjunction introduction. The real benefits

of reiteration come when we have multiple subproofs, as we will see in the following section (§29.1) – particularly when it comes to the negation rules. But we will also see later in §33.1 that, strictly speaking, we don't need the reiteration rule – though for convenience we will keep it.

Key Ideas in §28

- Natural deduction gives us rules that tell us how to infer *from* a sentence with a certain main connective (elimination rules), and rules that tell us how to infer *to* a sentence with a certain main connective (introduction rules).
- Proofs using the rules so far retain all the assumptions made during the course of the proof, and so correspond to an argument with those assumptions as premises and the final line of the proof as a conclusion.
- Reiteration is an optional rule, but useful for housekeeping in proofs.

Practice exercises

A. The following 'proof' is *incorrect*. Explain the mistakes it makes.

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

B. Give natural deduction proofs for the following arguments:

1. $P \therefore ((P \vee Q) \wedge P)$;
2. $((P \wedge Q) \wedge (R \wedge P)) \therefore ((R \wedge Q) \wedge P)$;
3. $(A \rightarrow (A \rightarrow B)), A \therefore B$;
4. $(B \leftrightarrow (A \leftrightarrow B)), B \therefore A$.

29

Basic Rules for Sentential: Rules with Subproofs

We've already seen in §27 how to start a proof by making assumptions. But the true power of natural deduction relies on its rules governing when you can make additional assumptions during the course of the proof, and how you can *discharge* those assumptions when you no longer need them.

29.1 Additional Assumptions and Subproofs

In natural deduction, both making and discharging additional assumptions are handled using SUBPROOFS. These are subsidiary proofs within the main proof, which encapsulate that part of a larger proof that depends on an assumption that is not among the premises. (Conversely, we can think of the premises of an argument as those assumptions left undischarged at the conclusion of a proof.)

When we start a subproof, we draw another vertical line (to the right of any existing assumption lines) to indicate that we are no longer in the main proof. Then we write in the assumption upon which the subproof will be based. A subproof can be thought of as essentially posing this question: *what could we show, if we also make this additional assumption?* We've already seen this in action earlier (§27), when we said that we could indicate the range of several premises in an argument by either attaching them all to one vertical assumption line, or introducing a new vertical line for each new assumption. In that case, we never got rid of the new assumptions: they remained as premises.

What will be new in this section is that some rules take us *back out* of a subproof. So the rules we will now consider are quite different from the rules covered in §28, none of which have this feature of being able to escape from a previously introduced assumption.

When we are working within a subproof, we can refer to the additional assumption that we made in introducing the subproof, and to anything that we obtained from our

original assumptions. (After all, those original assumptions are still in effect.) But at some point, we shall 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. This has been implicit in our discussion all along, but it is good to make it very clear:

> Any point in a natural deduction proof is in the range of some currently active assumptions, and the natural deduction rules can be applied only to sentences which do not rely on any other assumptions than those.
> Equivalently, any rule can be applied to any earlier lines in a proof, *except* for those lines which occur within a closed subproof.

Closing a subproof is called DISCHARGING the assumptions of that subproof. So we can put the point this way: *at no stage of a proof can you apply a rule to a sentence that occurs only in the range of an already discharged assumption*.

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, with those vertical assumption lines that indicate the range of an assumption. (Indeed, that's precisely why we have chosen to use *this* proof system.)

When can we begin a new subproof? *Whenever we want*. At any stage in a proof it is legitimate to assume something new, as long as we begin keeping track of what in our proof rests on this new assumption. No rule is needed to justify making an assumption, though there are guidelines that help us decide when it might be particularly appropriate to make a new assumption. Some of these are discussed in §32.

Even without any new rules, the idea of opening subproofs by making new assumptions can be used to illustrate a remark I made above about when a claim depends on assumptions (p. 240). Consider this proof:

1	$(P \wedge Q)$	
2	Q	$\wedge$E 1
3	$\quad R$	
4	$\quad Q$	R 2

In this proof, even though the occurrence of 'Q' on line 4 occurs within the range of the assumption 'R', it does not intuitively depend on it. We used reiteration to show that 'R' is still derivable from active assumptions at line 4, but it does not follow that 'R' depends in any robust way on every assumption that is active at line 4.

29.2 Conditional Introduction

To illustrate the use of subproofs, we will begin with the rule of conditional introduction. It is fairly easy to motivate informally. The following argument in English should be valid:

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

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

> Assume that Ludwig is reactionary. Now, *additionally* assume that Ludwig is libertarian. Then by conjunction introduction, it follows that Ludwig is both reactionary and libertarian. Of course, that only follows *conditional* on the assumption that Ludwig is libertarian. But this just means that, if Ludwig is libertarian, then he is both reactionary and libertarian – at least, that follows given our initial assumption that he is reactionary.

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

1	R

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

1	R
2	$\quad L$

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

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

1	R	
2	$\quad L$	
3	$\quad R \wedge L$	∧I 1, 2

The two vertical lines to the left of line 3 show that '$R \wedge L$' is in the range of both assumptions, and indeed depends on them collectively.

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

1	R	
2	$\quad L$	
3	$\quad R \wedge L$	$\wedge$I 1, 2
4	$L \to (R \wedge L)$	$\to$I 2–3

Observe that we have dropped back to using one vertical line. We are no longer relying on the additional assumption, 'L', since the conditional itself follows just from our original assumption, 'R'. The use of conditional introduction has discharged the temporary assumption, so that the final line of this proof relies only on the initial assumption 'R' – we made use of the assumption 'L' only in the nested subproof, and the range of that assumption is restricted to sentences in that subproof.

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 have established the following: If is does in fact turn out that A, then it also turns out that B. This is wrapped up in the rule for CONDITIONAL INTRODUCTION:

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

There can be as many or as few lines as you like between lines i and j. Notice that in our presentation of the rule, discharging the assumption $\mathcal{A}$ takes us out of the subproof in which $\mathcal{B}$ is derived from $\mathcal{A}$. If $\mathcal{A}$ is the initial assumption of a proof, then discharging it may well leave us with a conditional claim that depends on *no undischarged assumptions at all*. We see an example in this proof, where the main proof, marked by the leftmost vertical line, features no horizontal line marking an assumption:

1	$\quad P \wedge P$	
2	$\quad P$	$\wedge$E 1
3	$(P \wedge P) \to P$	$\to$I 1–2

It might come as no surprise that the conclusion of this proof – being provable from no undischarged assumptions at all – turns out to be a logical truth.

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

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

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

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 →E on the first premise. This will yield 'Q'. And we can then use →E on the second premise. So, by assuming 'P' we were able to prove 'R', so we apply the →I rule – discharging 'P' – and finish the proof. Putting all this together, we have:

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

Let's consider another example, this one demonstrating why reiteration can be so useful in subproofs. We know that $P \therefore Q \to P$ is a valid argument, from truth-tables. This is a proof:

1	P	
2	$\quad Q$	
3	$\quad P$	R 1
4	$(Q \to P)$	→I 2–3

Note that strictly speaking we needn't have used reiteration here: the assumption of 'P' remains active at line 2, so technically we could apply →I to close the subproof and

introduce the conditional immediately after line 2. But the use of reiteration makes it much clearer what is going on in the proof – even though all it does it repeat the earlier assumption and remind us that it is still an active assumption.

We now have all the rules we need to show that the argument on page 236 is valid. Here is the six line proof, some 175,000 times shorter than the corresponding truth table:

1	$A_1 \to C_1$	
2	$\quad (A_1 \land A_2 \land A_3 \land A_4 \land A_5 \land A_6 \land A_7 \land A_8 \land A_9 \land A_{10})$	
3	$\quad A_1$	$\land$E
4	$\quad C_1$	$\to$
5	$\quad (C_1 \lor C_2 \lor C_3 \lor C_4 \lor C_5 \lor C_6 \lor C_7 \lor C_8 \lor C_9 \lor C_{10})$	$\lor$I
6	$(A_1 \land A_2 \land A_3 \land A_4 \land A_5 \land A_6 \land A_7 \land A_8 \land A_9 \land A_{10}) \to (C_1 \lor C_2 \lor C_3 \lor C_4 \lor C_5 \lor C_6 \lor C_7 \lor C_8 \lor C_9 \lor C_{10})$	$\to$

Import-Export

Our rules so far can be used to demonstrate two important and rather controversial principles governing the conditional. The principle of IMPORTATION is the claim that from 'if P then, if also Q, then R' it follows that 'if P and also Q, then R'. The principle of EXPORTATION is the converse, that from 'if P and also Q, then R' it follows that 'if P, then if also Q, then R'. First, we prove importation holds for our conditional:

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

Second, we show exportation holds. Here, we need to open two nested subproofs:

1	$((P \wedge Q) \rightarrow R)$	
2	$\quad P$	
3	$\quad\quad Q$	
4	$\quad\quad (P \wedge Q)$	$\wedge$I 2, 3
5	$\quad\quad R$	$\rightarrow$E 1, 4
6	$\quad (Q \rightarrow R)$	$\rightarrow$I 3–5
7	$(P \rightarrow (Q \rightarrow R))$	$\rightarrow$I 2–6

29.3 Some Pitfalls of Additional Assumptions

Making additional assumptions in the range of an assumption needs to be handled with some care, as I said in §29.1. Now that we have a rule that discharges assumptions in our repertoire, we can describe some of the potential pitfalls.

Consider this proof:

1	A	
2	$\quad B$	
3	$\quad B \wedge B$	$\wedge$I 2, 2
4	$\quad B$	$\wedge$E 3
5	$B \rightarrow B$	$\rightarrow$I 2–4

This is perfectly in keeping with the rules we have laid down already. And it should not seem particularly strange. Since '$B \rightarrow B$' is a logical truth, no particular premises should be required to prove it – note that 'A' plays no particular role in the proof apart from beginning 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	$B \rightarrow B$	$\rightarrow$I 2–4
6	B	naughty attempt to invoke $\rightarrow$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 Anne is fast (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. The rule on page 256 stipulation rules out the disastrous attempted proof above. The rule of →E requires that we cite two individual lines from earlier in the proof. In the purported proof, above, one of these lines (namely, line 4) occurs within a subproof that has (by line 6) been closed. This is illegitimate.

29.4 Subproofs within Subproofs

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

1	A	
2	$\quad B$	
3	$\quad\quad C$	
4	$\quad\quad A \wedge B$	∧I 1, 2
5	$\quad C \rightarrow (A \wedge B)$	→I 3–4
6	$B \rightarrow (C \rightarrow (A \wedge B))$	→I 2–5

Notice that the commentary 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	$\quad\quad A \wedge B$	∧I 1, 2
5	$\quad C \rightarrow (A \wedge B)$	→I 3–4
6	$B \rightarrow (C \rightarrow (A \wedge B))$	→I 2–5
7	$C \rightarrow (A \wedge B)$	naughty attempt to invoke →I 3–4

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

The essential problem is that the subproof that began with the assumption 'C' depended crucially on the fact that we had assumed 'B' on line 2. By line 6, we have *discharged* the assumption 'B': we have stopped asking ourselves what we could show, if we also assumed 'B'. So it is simply cheating, to try to help ourselves (on line 7) to the subproof that began with the assumption 'C'. The attempted disastrous proof violates, as before, the rule in the box on page 256. The subproof of lines 3–4 occurs within a subproof that ends on line 5. Its assumptions are discharged before line 7, so they cannot be invoked in any rule which applies to produce line 7.

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

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

Recall the proof of the argument

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

from page 259. One thing to note about the proof there is that because there are two assumptions with the same range in the main proof, it is not easily possible to discharge just one of them using the →I rule. For that rule only applies to a one-assumption subproof. If we wanted to discharge another of our assumptions, we shall have to put the proof into the right form, with each assumption made individually as the head of its own subproof:

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

The conclusion is now in the range of both assumptions, as in the earlier proof – but now it is also possible to discharge these assumptions if we wish:

1	$P \to Q$	
2	$\quad Q \to R$	
3	$\quad\quad P$	
4	$\quad\quad Q$	$\to$E 1, 3
5	$\quad\quad R$	$\to$E 2, 4
6	$\quad P \to R$	$\to$I 3–5
7	$(Q \to R) \to (P \to R)$	$\to$I 2–6
8	$(P \to Q) \to ((Q \to R) \to (P \to R))$	$\to$I 1–7

While it is permissible, and often convenient, to have several assumptions with the same range and without nesting, I recommend always trying to construct your proofs so that each assumption begins its own subproof. That way, if you later wish to apply rules which discharge a single assumption, you may always do so.

29.5 Proofs within Proofs

One interesting feature of a natural deduction system like ours is that because we can make any assumption at any point, and thereafter continue in accordance with the rules, any correctly formed proof can be *re-used* as a subproof in a later proof. For example, suppose we wanted to give a proof of this argument:

$$\big((P \to Q) \land (P \to R)\big) \therefore (P \to (R \land Q)).$$

We begin by opening our proof by assuming the premise. We also note that the conclusion is a conditional, and so we'll assume that it is obtained by an instance of conditional introduction. That will give us this 'skeleton' of a proof before we begin filling in the details:

1	$\big((P \to Q) \land (P \to R)\big)$	
2	$\quad P$	
	$\quad \vdots$	
m	$\quad (R \land Q)$	$\land$I 6, 5
$m+1$	$(P \to (R \land Q))$	$\to$I 2–m

Then we recall – perhaps! – that we already have a proof that looks very much like this. On page 250 we have a proof that uses the same premise as ours, but also uses the premise 'P' to derive '$(R \land Q)$' – which is what we need. So we can simply copy that whole proof over to fill in the missing section of our proof:

1	$\big((P \to Q) \land (P \to R)\big)$	
2	$\quad P$	
3	$\quad (P \to Q)$	$\land$E 1
4	$\quad (P \to R)$	$\land$E 1
5	$\quad Q$	$\to$E 3, 2
6	$\quad R$	$\to$E 4, 2
7	$\quad (R \land Q)$	$\land$I 6, 5
8	$(P \to (R \land Q))$	$\to$I 2–7

This is a very useful feature: for if you have proved something once, you can re-use that proof whenever you need to, as a subproof in some other proof.

The converse isn't always true, because sometimes in a subproof you use an assumption from outside the subproof, and if you don't make the same assumption in your other proof, the displaced subproof may no longer correctly follow all the rules it uses.

29.6 Biconditional Introduction

The biconditional is like a two-way conditional. The introduction rule for the biconditional resembles two instances of conditional introduction, one for each direction.

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 to license the introduction. Schematically, the rule works like this:

i	$\quad \mathcal{A}$	
	$\quad \vdots$	
j	$\quad \mathcal{B}$	
k	$\quad \mathcal{B}$	
	$\quad \vdots$	
l	$\quad \mathcal{A}$	
	$\vdots$	
	$(\mathcal{A} \leftrightarrow \mathcal{B})$	$\leftrightarrow$I i–j, k–l

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. Again, this rule permits us to discharge assumptions, and the same restrictions on making use of claims derived in a closed subproof outside of that subproof apply.

We can now prove that a biconditional is like two conjoined conditionals. Using the conditional and biconditional rules, we can prove that a biconditional entails a conjunction of conditionals, and *vice versa*:

1	$(A \leftrightarrow B)$	
2	$\quad A$	
3	$\quad B$	$\leftrightarrow$E 1, 2
4	$(A \rightarrow B)$	$\rightarrow$I 2–3
5	$\quad B$	
6	$\quad A$	$\leftrightarrow$E 1, 5
7	$(B \rightarrow A)$	$\rightarrow$I 5–6
8	$\big((A \rightarrow B) \wedge (B \rightarrow A)\big)$	$\wedge$I 4, 7

1	$\big((A \rightarrow B) \wedge (B \rightarrow A)\big)$	
2	$(A \rightarrow B)$	$\wedge$E 1
3	$\quad A$	
4	$\quad B$	$\rightarrow$E 2, 3
5	$(B \rightarrow A)$	$\wedge$E 1
6	$\quad B$	
7	$\quad A$	$\rightarrow$E 5, 6
8	$(A \leftrightarrow B)$	$\leftrightarrow$I 3–4, 6–7

Another informative example demonstrates the logical equivalence of '$((P \wedge Q) \rightarrow R)$' and '$(P \rightarrow (Q \rightarrow R))$' given importation and exportation (page 260). We will re-use both of our earlier proofs, stitching them together using biconditional introduction in the final line:

1	$(P \to (Q \to R))$	
2	$\quad (P \land Q)$	
3	$\quad P$	$\land$E 2
4	$\quad (Q \to R)$	$\to$E 1, 3
5	$\quad Q$	$\land$E 2
6	$\quad R$	$\to$E 4, 5
7	$((P \land Q) \to R)$	$\to$I 2–6
8	$((P \land Q) \to R)$	
9	$\quad P$	
10	$\quad\quad Q$	
11	$\quad\quad (P \land Q)$	$\land$I 9, 10
12	$\quad\quad R$	$\to$E 8, 11
13	$\quad (Q \to R)$	$\to$I 10–12
14	$(P \to (Q \to R))$	$\to$I 9–13
15	$\big((P \land Q) \to R) \leftrightarrow (P \to (Q \to R))\big)$	$\leftrightarrow$I 8–14, 1–7

Note the small gap between the nested vertical lines between lines 7 and 8 – that shows we have two subproofs here, not one. (That would also be indicated by the fact that the sentence on line 8 has a horizontal line under it – no vertical assumption line has two markers of where the assumptions cease.)

The acceptability of our proof rules is grounded in the fact that they will never lead us from truth to falsehood. The acceptability of the biconditional introduction rule is demonstrated by the following correct entailment:

› If $\mathcal{C}_1, \ldots, \mathcal{C}_n, \mathcal{A} \vDash \mathcal{B}$ and $\mathcal{C}_1, \ldots, \mathcal{C}_n, \mathcal{B} \vDash \mathcal{A}$, then $\mathcal{C}_1, \ldots, \mathcal{C}_n \vDash \mathcal{A} \leftrightarrow \mathcal{B}$.

29.7 Disjunction Elimination

The disjunction elimination rule is slightly trickier than those we've seen so far. Suppose that either Ludwig is reactionary or he is libertarian. What can you conclude? Not that Ludwig is reactionary; it might be that he is libertarian instead. And equally, not that Ludwig is libertarian; for he might merely be reactionary. It can be hard to draw a definite conclusion from a disjunction just by itself.

But suppose that we could somehow show both of the following: first, that Ludwig's being reactionary entails that he is an Austrian economist: second, that Ludwig's being libertarian *also* entails that he is an Austrian economist. Then if we know that Ludwig

is either reactionary or libertarian, then we know that, whichever he is, Ludwig is an Austrian economist. This we might call 'no matter whether' reasoning: if each of $\mathcal{A}$ and $\mathcal{B}$ imply $\mathcal{C}$, then *no matter whether* $\mathcal{A}$ *or* $\mathcal{B}$, still $\mathcal{C}$. Sometimes this kind of reasoning is called *proof by cases*, since you start with the assumption that either of two cases holds, and then show something follows no matter which case is actual.

This insight can be expressed in the following rule, which is our DISJUNCTION ELIMINATION (∨E) rule:

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

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

An adaptation of the previous proof can be used to establish a proof for this argument:

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

We begin the cases in the same way as above, but as we continue please note the use of the disjunction introduction rule to get the last line of each subproof in the right format to use disjunction elimination.

1	$(P \wedge Q) \vee (P \wedge R)$	
2	$\quad P \wedge Q$	
3	$\quad P$	$\wedge$E 2
4	$\quad Q$	$\wedge$E 2
5	$\quad (Q \vee R)$	$\vee$I 4
6	$\quad P \wedge (Q \vee R)$	$\wedge$I 3, 5
7	$\quad P \wedge R$	
8	$\quad P$	$\wedge$E 7
9	$\quad R$	$\wedge$E 7
10	$\quad (Q \vee R)$	$\vee$I 9
11	$\quad P \wedge (Q \vee R)$	$\wedge$I 8, 10
12	$P \wedge (Q \vee R)$	$\vee$E 1, 2–6, 7–11

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

Another slightly tricky example. Consider:

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

Here is a proof corresponding to this argument:

1	$A \land (B \lor C)$	
2	A	$\land$E 1
3	$B \lor C$	$\land$E 1
4	$\quad B$	
5	$\quad A \land B$	$\land$I 2, 4
6	$\quad (A \land B) \lor (A \land C)$	$\lor$I 5
7	$\quad C$	
8	$\quad A \land C$	$\land$I 2, 7
9	$\quad (A \land B) \lor (A \land C)$	$\lor$I 8
10	$(A \land B) \lor (A \land C)$	$\lor$E 3, 4–6, 7–9

This disjunction rule is supported by the following valid Sentential argument form:

› If $\mathcal{D}_1, \ldots, \mathcal{D}_n, \mathcal{A} \lor \mathcal{B}, \mathcal{A} \vDash \mathcal{C}$ and $\mathcal{D}_1, \ldots, \mathcal{D}_n, \mathcal{A} \lor \mathcal{B}, \mathcal{B} \vDash \mathcal{C}$, then $\mathcal{D}_1, \ldots, \mathcal{D}_n, \mathcal{A} \lor \mathcal{B} \vDash \mathcal{C}$.

29.8 Negation Introduction

Our negation rules are inspired by the form of reasoning known as *reductio* (recall page 230). In *reductio* reasoning, we make an assumption that $\mathcal{A}$ for the sake of argument, and show that something contradictory follows from it. Then we can conclude that our assumption was false, and that its negation must be true. There are lots of approaches to negation in natural deduction, but all of them stem from this same basic insight: when an assumption that $\mathcal{A}$ goes awry, conclude $\neg\mathcal{A}$.

We see *reductio* reasoning used a lot in mathematics. For example: Suppose there is a largest number, call it n. Since n is the largest number, $n + 1 \leqslant n$. But then, subtracting n from both sides, $1 \leqslant 0$. And that is absurd. So there is no largest number. Here, we make an assumption, for the sake of argument. We derive from it a claim, in this case that $1 \leqslant 0$. And we note that claim is absurd, *given what we already know*. So we conclude that the negation of our assumption holds, and cease to rely on the problematic assumption.

The only claims in logic that it is safe to say are absurd are logical falsehoods. So in a logical version of *reductio* reasoning, we will want to show that claims that contradict one another will be derivable in the range of an assumption, in order to prove the negation of that assumption. Our NEGATION INTRODUCTION rule fits this pattern very clearly:

i	$\quad\mid\ \mathcal{A}$	
	$\quad\mid\ \vdots$	
j	$\quad\mid\ \mathcal{B}$	
	$\quad\mid\ \vdots$	
k	$\quad\mid\ \neg\mathcal{B}$	
	$\neg\mathcal{A}$	$\neg$I i–j, i–k

Here, we can prove a sentence and its negation both within the range of the assumption that $\mathcal{A}$. So if $\mathcal{A}$ were assumed, something contradictory would be derivable under that assumption. (We could apply conjunction introduction to lines j and k to make the logical falsehood explicit, but that wouldn't be strictly necessary.) Since logical falsehoods are fundamentally unacceptable as the termination of a chain of argument, we must have begun with an inappropriate starting point when we assumed $\mathcal{A}$. So, in fact, we conclude, $\neg\mathcal{A}$, discharging our erroneous assumption that $\mathcal{A}$. There is no need for the line with $\mathcal{B}$ on it to occur before the line with $\neg\mathcal{B}$ on it.

Almost always the logical falsehood arises because of a clash between the claim we assume and some bit of prior knowledge – typically, some claim we have established earlier in the proof. We will thus make frequent use of the rule of reiteration in applications of negation introduction, to get the contradictory claims in the right place to make the rule easy to apply. Here is an example of the rule in action, showing that this argument is provable:

$$A, \neg B \therefore \neg(A \to B).$$

1	A	
2	$\neg B$	
3	$\quad\mid\ A \to B$	
4	$\quad\mid\ B$	$\to$E 3, 1
5	$\quad\mid\ \neg B$	R 2
6	$\neg(A \to B)$	$\neg$I 3–4, 3–5

Another example, for practice. Let's prove this argument:

$$(C \to \neg A) \therefore (A \to \neg C).$$

1	$(C \to \neg A)$	
2	$\quad A$	
3	$\quad\quad C$	
4	$\quad\quad \neg A$	$\to$E 1, 3
5	$\quad\quad A$	R 2
6	$\quad \neg C$	$\neg$I 3–5, 3–4
7	$(A \to \neg C)$	$\to$I 2–6

The correctness of the negation introduction rule is demonstrated by this valid Sentential argument form:

› If $\mathcal{C}_1, \ldots, \mathcal{C}_n, \mathcal{A} \vDash \mathcal{B}$ and $\mathcal{C}_1, \ldots, \mathcal{C}_n, \mathcal{A} \vDash \neg\mathcal{B}$, then $\mathcal{C}_1, \ldots, \mathcal{C}_n \vDash \neg\mathcal{A}$.

29.9 Negation Elimination

The rule of negation introduction is interesting, because it is *almost* its own elimination rule too! Consider this schematic proof:

i	$\quad \neg\mathcal{A}$	
	$\quad \vdots$	
j	$\quad \mathcal{B}$	
	$\quad \vdots$	
k	$\quad \neg\mathcal{B}$	
$k+1$	$\neg\neg\mathcal{A}$	$\neg$I i–j, i–k

This proof terminates with a sentence that is logically equivalent to $\mathcal{A}$, discharging the assumption that $\neg\mathcal{A}$ because it leads to contradictory conclusions. This looks awfully close to a rule of negation elimination – if only we could find a way to replace a doubly-negated sentence $\neg\neg\mathcal{A}$ by the logically equivalent sentence $\mathcal{A}$, which would have eliminated the negation from the problematic assumption $\neg\mathcal{A}$.

In our system, we approach this problem by the brute force method – we allow ourselves to use the derivation of contradictory sentences from a negated sentence to motivate the elimination of that negation. This leads to our rule of NEGATION ELIMINATION:

i	$\quad\neg\mathcal{A}$	
	$\quad\vdots$	
j	$\quad\mathcal{B}$	
	$\quad\vdots$	
k	$\quad\neg\mathcal{B}$	
	$\mathcal{A}$	$\neg$E i–j, i–k

This is also *reductio* reasoning, though in this case from a negated assumption. But again, if the assumption of $\neg\mathcal{A}$ goes awry and allows us to derive contradictory claims (perhaps given what we've already shown), that licenses us to conclude $\mathcal{A}$.

With the rule of negation elimination, we can prove some claims that are hard to prove directly. For example, suppose we wanted to prove the law of EXCLUDED MIDDLE, that '$(P \vee \neg P)$'. You might initially have thought: *it is a disjunction, so should be proved by disjunction introduction*. But we cannot prove either the sentence letter 'P' or its negation from no assumptions – so we could not prove excluded middle from no assumptions if it was by disjunction introduction from one of its disjuncts. So we proceed indirectly: we show that supposing the negation of the law of excluded middle leads to logical falsehood, and conclude it by negation elimination:

1	$\quad\neg(P \vee \neg P)$	
2	$\quad\quad P$	
3	$\quad\quad (P \vee \neg P)$	$\vee$I 2
4	$\quad\quad \neg(P \vee \neg P)$	R 1
5	$\quad\neg P$	$\neg$I 2–3, 2–4
6	$\quad(P \vee \neg P)$	$\vee$I 5
7	$\quad\neg(P \vee \neg P)$	R 1
8	$(P \vee \neg P)$	$\neg$E 1–6, 1–7

One interesting feature of this proof is that one of the contradictory sentences is the assumption itself. When the assumption that $\neg\mathcal{A}$ goes wrong, it might be because we have the resources to prove $\mathcal{A}$!

To see our negation rules in action, consider:

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

Here is a proof corresponding with the argument:

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

I make two comments. In line 6, the justification cites line 2 which lies outside the subproof. That is okay, since the application of the rule lies within the range of the assumption of line 2. In line 9, the justification only cites the subproof from 2 to 8, rather than two ranges of line numbers. This is because in this application of our rule, we have the special case where the sentence such that both it and its negation can be derived from the assumption *is* that assumption. It would be trivial to derive it from itself.

The negation elimination rule is supported by this valid Sentential argument form:

› If $\mathcal{C}_1, \ldots, \mathcal{C}_n, \neg\mathcal{A} \vDash \mathcal{B}$ and $\mathcal{C}_1, \ldots, \mathcal{C}_n, \neg\mathcal{A} \vDash \neg\mathcal{B}$, then $\mathcal{C}_1, \ldots, \mathcal{C}_n \vDash \mathcal{A}$;

29.10 Putting it All Together

We have now explained all of the basic rules for the proof system for Sentential. Let's return to some of the arguments from §27 with which we began our exploration of this system, to see how they can be proved. And I will give a third example of a complex proof that uses many of our rules.

1. One argument we considered earlier was this:

$$\neg(A \lor B) \therefore (\neg A \land \neg B).$$

   We can now see that the proof we began to construct can be completed can be proved as in Figure 29.1.

2. The second proof we began constructing earlier corresponded to this argument:

$$(A \lor B), \neg(A \land C), \neg(B \land \neg D) \therefore (\neg C \lor D).$$

   The proof can be completed as in Figure 29.2.

3. Finally, Figure 29.3 shows a long proof involving most of our rules in action.

1	$\neg(A \vee B)$	
2	$\quad A$	
3	$\quad (A \vee B)$	$\vee$I 2
4	$\quad \neg(A \vee B)$	R 1
5	$\neg A$	$\neg$I 2–3, 2–4
6	$\quad B$	
7	$\quad (A \vee B)$	$\vee$I 6
8	$\quad \neg(A \vee B)$	R 1
9	$\neg B$	$\neg$I 6–7, 6–8
10	$(\neg A \wedge \neg B)$	$\wedge$I 5, 9

Figure 29.1: Proof of $\neg(A \vee B) \therefore (\neg A \wedge \neg B)$

These three proofs are more complex than the others we've considered, because they involve multiple rules in tandem. You should make sure you understand why each rule applies where it does, and that the proofs are correct, before you move on. You probably won't feel that you are able to construct a proof yourself as yet, and that is okay. It is important now to see that these are in fact proofs. Some ideas about how to go about constructing them yourself will be presented in §32. But you will also get a sense about how to construct complex proofs as you practice constructing simpler proofs and start to see how they can be slotted together to form larger proofs. There is no substitute for practice.

Key Ideas in §29

- The rules for our system are summarised on page 348.
- It is important that we keep track of restrictions on when we can make use of claims derived in a subproof, since those subproofs may be making use of assumptions we are no longer accepting.
- Our proof rules match the interpretation of Sentential we have given – they will not permit us to say that some claim is provable from some assumptions when that claim isn't entailed by those assumptions.

1	$A \vee B$	
2	$\quad \neg(A \wedge C)$	
3	$\quad\quad \neg(B \wedge \neg D)$	
4	$\quad\quad\quad A$	
5	$\quad\quad\quad\quad C$	
6	$\quad\quad\quad\quad A \wedge C$	$\wedge$I 4, 5
7	$\quad\quad\quad\quad \neg(A \wedge C)$	R 2
8	$\quad\quad\quad \neg C$	$\neg$I 5–6, 5–7
9	$\quad\quad\quad (\neg C \vee D)$	$\vee$I 8
10	$\quad\quad\quad B$	
11	$\quad\quad\quad\quad \neg D$	
12	$\quad\quad\quad\quad (B \wedge \neg D)$	$\wedge$I 10, 11
13	$\quad\quad\quad\quad \neg(B \wedge \neg D)$	R 3
14	$\quad\quad\quad D$	$\neg$E 11–12, 11–13
15	$\quad\quad\quad (\neg C \vee D)$	$\vee$I 14
16	$\quad\quad (\neg C \vee D)$	$\vee$E 1, 4–9, 10–15

Figure 29.2: Proof that $(A \vee B), \neg(A \wedge C), \neg(B \wedge \neg D) \therefore (\neg C \vee D)$.

Practice exercises

A. The following 'proof' is *incorrect*. Explain the mistakes it makes.

1	$\neg P \lor Q$	
2	$\quad \neg P$	
3	$\quad\quad P$	
4	$\quad\quad\quad \neg Q$	
5	$\quad\quad\quad P \land \neg P$	$\land$I 3, 2
6	$\quad\quad\quad P$	$\land$E 5
7	$\quad\quad\quad \neg P$	$\land$E 5
8	$\quad\quad Q$	$\neg$E 4–7
9	$\quad P \to Q$	$\to$I 3–8
10	$\quad Q$	
11	$\quad\quad P$	
12	$\quad\quad Q \land Q$	$\land$I 10, 10
13	$\quad\quad Q$	$\land$E 12
14	$\quad P \to Q$	$\to$I 11–13
15	$P \to Q$	$\lor$E 1, 2–9, 10–14
16	$P \to Q$	
17	$\quad \neg(\neg P \lor Q)$	
18	$\quad\quad P$	
19	$\quad\quad Q$	$\to$E 16, 18
20	$\quad\quad \neg P \lor Q$	$\lor$I 19
21	$\quad \neg P$	$\neg$I 18–17, 18–20
22	$\quad \neg P \lor Q$	$\lor$I 21
23	$\neg P \lor Q$	$\neg$E 17–17, 17–22
24	$((\neg P \lor Q) \leftrightarrow (P \to Q))$	$\leftrightarrow$I 1–15, 16–23

Figure 29.3: A complicated proof

1	$\neg L \to (A \land L)$	
2	$\quad \neg L$	
3	$\quad A$	$\to$E 1, 2
4	$\quad L$	
5	$\quad L \land \neg L$	$\land$I 4, 2
6	$\quad\quad \neg A$	
7	$\quad\quad L$	$\to$I 5
8	$\quad\quad \neg L$	$\to$E 5
9	$\quad A$	$\neg$I 6–7, 6–8
10	A	$\lor$E 2–3, 4–9

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

1	$P \land S$
2	P
3	S
4	$\quad S \to R$
5	$\quad R$
6	$\quad R \lor E$

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

1	$\neg L \to (J \lor L)$
2	$\quad \neg L$
3	$\quad J \lor L$
4	$\quad\quad J$
5	$\quad\quad J \land J$
6	$\quad\quad J$
7	$\quad\quad L$
8	$\quad\quad\quad \neg J$
9	$\quad\quad J$
10	$\quad J$

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

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

30

Some Philosophical Issues about Conditionals and Meaning

30.1 Conditional Introduction and the English Conditional

We motivated the conditional introduction rule back on page 257 by giving a English argument, using the English word 'if'. Now, →I is a stipulated rule for our conditional connective →; it doesn't really need motivation since we could simply postulate that such a rule is part of our formal proof system. It is justified, if justification is needed, by the Deduction Theorem (a result noted in §11.3). (Likewise, →E may be justified by a schematic truth table demonstration that $\mathcal{A}, \mathcal{A} \to \mathcal{C} \vDash \mathcal{C}$.)

But if we are to offer a motivation for our rule in English, then we must be relying on the plausibility of this English analog of →I, known as CONDITIONAL PROOF:

> If you can establish $\mathcal{C}$, given the assumption that $\mathcal{A}$ and perhaps some supplementary assumptions $\mathcal{B}_1, \ldots, \mathcal{B}_n$, then you can establish 'if $\mathcal{A}$, $\mathcal{C}$' solely on the basis of the assumptions $\mathcal{B}_1, \ldots, \mathcal{B}_n$.

Conditional proof captures a significant aspect of the English 'if': the way that conditional helps us neatly summarise reasoning from assumptions, and then store that reasoning for later use in a conditional form.

But if conditional proof is a good rule for English 'if', then we can argue that 'if' is actually synonymous with '→':

> Suppose that $\mathcal{A} \to \mathcal{C}$. Assume $\mathcal{A}$. We can now derive $\mathcal{C}$, by →E. But we have now established $\mathcal{C}$ on the basis of the assumption $\mathcal{A}$, together with the supplementary assumption $\mathcal{A} \to \mathcal{C}$. By conditional proof, then, we can establish 'if $\mathcal{A}$, $\mathcal{C}$' on the basis of the supplementary assumption $\mathcal{A} \to \mathcal{C}$ alone. But that of course means that we can derive (in English) an English conditional sentence from a Sentential conditional sentence, with the

appropriate interpretation of the constituents $\mathcal{A}$ and $\mathcal{C}$. Since the English conditional sentence obviously suffices for the Sentential conditional sentence, we have shown them to be synonymous.

Our discussion in §11.5 seemed to indicate that the English conditional was not synonymous with '→'. That is hard to reconcile with the above argument. Most philosophers have concluded that, contrary to appearances, conditional proof is not always a good way of reasoning for English 'if'. Here is an example which seems to show this, though it requires some background.

It is clearly valid in English, though rather pointless, to argue from a claim to itself. So when $\mathcal{C}$ is some English sentence, $\mathcal{C} \therefore \mathcal{C}$ is a valid argument in English. And we cannot make a valid argument invalid by adding more premises: if premises we already have are conclusive grounds for the conclusion, adding more premises while keeping the conclusive grounds cannot make the argument less conclusive. So $\mathcal{C}, \mathcal{A} \therefore \mathcal{C}$ is a valid argument in English.

If conditional proof were a good way of arguing in English, we could convert this valid argument into another valid argument with this form: $\mathcal{C} \therefore$ if $\mathcal{A}, \mathcal{C}$. But then conditional proof would then allow us to convert this valid argument:

174. I will go skiing tomorrow;
175. I break my leg tonight;
So: I will go skiing tomorrow. (which is just repeating 174 again)

into this intuitively invalid argument:

174. I will go skiing tomorrow;
So: If I break my leg tonight, I will go skiing tomorrow.

This is invalid, because even if the premise is true, the conclusion seems to be actually false. If conditional proof enables us to convert a valid English argument into an invalid argument, so much the worse for conditional proof.

There is much more to be said about this example. What is beyond doubt is that →I is a good rule for Sentential, regardless of the fortunes of conditional proof in English. It does seem that instances of conditional proof in mathematical reasoning are all acceptable, which again shows the roots of natural deduction as a formalisation of existing mathematical practice. This would suggest that → might be a good representation of mathematical uses of 'if'.

30.2 Inferentialism

You will have noticed that our rules come in pairs: an introduction rule that tells you how to introduce a connective into a proof from what you have already, and an elimination rule that tells you how to remove it in favour of its consequences. These rules can be justified by consideration of the meanings we assigned to the connectives of Sentential in the schematic truth tables of §8.3.

But perhaps we should invert this order of justification. After all, the proof rules already seem to capture (more or less) how 'and', 'or', 'not', 'if', and 'iff' work in conversation. We make some claims and assumptions. The introduction and elimination rules summarise how we might proceed in our conversation against the background of those claims and assumptions. Many philosophers have thought that the meaning of an expression is entirely governed by how it is or might be used in a conversation by competent speakers – in slogan form, *meaning is fixed by use*. If these proof rules describe how we might bring an expression into a conversation, and what we may do with it once it is there, then these proof rules describe the totality of facts on which meaning depends. The meaning of a connective, according to this INFERENTIALIST picture, is represented by its introduction and elimination rules – and not by the truth-function that a schematic truth table represents. On this view, it is the correctness of the schematic proof of $\mathcal{A} \vee \mathcal{B}$ from $\mathcal{A}$ which explains *why* the schematic truth table for '$\vee$' has a T on every row on which at least one of its constituents gets a T.

There is a significant debate on just this issue in the philosophy of language, about the nature of *meaning*. Is the meaning of a word what it represents, the view sometimes called REPRESENTATIONALISM? Or is the meaning of a word, rather, given by some rules for how to use it, as inferentialism says? We cannot go deeply into this issue here, but I will say a little. The representationalist view seems to accomodate some expressions very well: the meaning of a name, for example, seems very plausibly to be identified with what it names; the meaning of a predicate might be thought of as the corresponding property. But inferentialism seems more natural as an approach to the logical connectives:

> Anyone who has learnt to perform [conjunction introduction and conjunction elimination] knows the meaning of 'and', for there is simply nothing more to knowing the meaning of 'and' than being able to perform these inferences.[1]

It seems rather unnatural, by contrast, to think that the meaning of 'and' is some abstract mathematical 'thing' represented by a truth table.

Can the inferentialist distinguish good systems of rules, such as those governing 'and', from bad systems? The problem is that without appealing to truth tables or the like, we seem to be committed to the legitimacy of rather problematic connectives. The most famous example is Prior's 'tonk' governed by these rules:

m	$\mathcal{A}$		m	$\mathcal{A}$ tonk $\mathcal{B}$	
	$\vdots$			$\vdots$	
n	$\mathcal{A}$ tonk $\mathcal{B}$	tonk-I m	n	$\mathcal{B}$	tonk-E m

You will notice that 'tonk' has an introduction rule like '$\vee$', and an elimination rule like '$\wedge$'. Of course 'tonk' is a connective we would not like in a language, since pairing the introduction and elimination rules would allow us to prove any arbitrary sentence from any assumption whatsover:

[1] A N Prior (1961) 'The Runabout Inference-Ticket', *Analysis* **21**, p. 38.

1	P	
2	P tonk Q	tonk-I 1
3	Q	tonk-E 2

If we are to rule out such deviant connectives as 'tonk', Prior argues, we have to accept that 'an expression must have some independently determined meaning before we can discover whether inferences involving it are valid or invalid' (Prior, *op. cit.*, p. 38). We cannot, that is, accept the inferentialist position that the rules of implication come first and the meaning comes second. Inferentialists have replied, but we must unfortunately leave this interesting debate here for now.[2]

Key Ideas in §30

- The question of how to understand conditionals in natural language is a tricky one. The natural deduction rules we adopt are suitable to understand the logical conditional '→', but this may only be an approximation of English 'if'.
- The philosophical question of whether connectives are given meaning by their truth tables or by their natural deduction rules is an interesting one.

[2] The interested reader might wish to start with this reply to Prior: Nuel D Belnap, Jr (1962) 'Tonk, Plonk and Plink', *Analysis* **22**, pp. 130–34.

31

Proof-Theoretic Concepts

We shall introduce some new vocabulary. The following expression:

$$\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{C}$$

means that there is some proof which starts with assumptions among $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ and ends with $\mathcal{C}$ (and no undischarged assumptions other than those we started with). We can read this claim as '$\mathcal{C}$ is provable from $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$'.

The symbol '$\vdash$' is called the *single turnstile*. I want to emphasise that this is not the double turnstile symbol ('$\vDash$') that we used to symbolise entailment in chapters 3 and 5. The single turnstile, '$\vdash$', concerns the existence of formal proofs; the double turnstile, '$\vDash$', concerns the existence of valuations (or interpretations, when used for Quantifier). *They are very different notions.*[1]

Derivatively, we shall write:

$$\vdash \mathcal{A}$$

to mean that there is a proof of $\mathcal{A}$ which ends up having no undischarged assumptions. We now define:

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

To illustrate this, suppose I want to prove that '$\neg(A \wedge \neg A)$' is a theorem. So I must start my proof without *any* assumptions. However, since I want to prove a sentence whose main connective is a negation, I shall want to immediately begin a subproof by making the additional assumption '$A \wedge \neg A$' for the sake of argument, and show that this leads to contradictory consequences. All told, then, the proof looks like this:

[1] They are very different, but if we've designed our proof system well, we shouldn't be able to prove a conclusion from some assumptions *unless* that conclusion validly follows from those assumptions. And if we are really fortunate, we should be able to provide a proof corresponding to any valid argument. More on this in §38.

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

We have therefore proved '$\neg(A \wedge \neg A)$' on no (undischarged) assumptions. This particular theorem is an instance of what is sometimes called *the Law of Non-Contradiction*, that for any $\mathcal{A}$, $\neg(\mathcal{A} \wedge \neg\mathcal{A})$. We actually already saw a proof showing that the law of excluded middle is a theorem in §29.9.

To show that something is a theorem, you just have to find a suitable proof. It is typically much harder to show that something is *not* a theorem. To do this, you would have to demonstrate, not just that certain proof strategies fail, but that *no* proof is possible. Even if you fail in trying to prove a sentence in a thousand different ways, perhaps the proof is just too long and complex for you to make out. Perhaps you just didn't try hard enough.

Here is another new bit of terminology:

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

As in the case of showing that a sentence is a theorem, it is relatively easy to show that two sentences are provably equivalent: it just requires a pair of proofs. Showing that sentences are *not* provably equivalent would be much harder: it is just as hard as showing that a sentence is not a theorem. Notice that if we have two proofs, one demonstrating $\mathcal{A} \vdash \mathcal{B}$ and the other demonstrating $\mathcal{B} \vdash \mathcal{A}$, we can put them together as subproofs of a single proof, and apply $\leftrightarrow$I to provide a proof from no assumptions which demonstrates that $\vdash \mathcal{A} \leftrightarrow \mathcal{B}$.

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 logical falsehood can be proved from them, i.e., for some $\mathcal{B}$, $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{B} \wedge \neg\mathcal{B}$.

It is easy to show that some sentences are jointly contrary: you just need to provide a proof of a logical falsehood from assuming all the sentences. Showing that some sentences are not jointly contrary is much harder. It would require more than just providing a proof or two; it would require showing that no proof of a certain kind is *possible*.

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

	Yes	No
theorem?	one proof	all possible proofs
equivalent?	two proofs	all possible proofs
not contrary?	all possible proofs	one proof
provable	one proof	all possible proofs

Key Ideas in §31

- Our formal proof system allows us to introduce a notion of provability, symbolised '$\vdash$'. This is distinct from entailment '$\vDash$', but (if we've done our work correctly) they will parallel one another.
- We can introduce further notions in terms of provability: provable equivalence, joint contrariness, and being a theorem.

Practice exercises

A. Give a proof showing that each of the following sentences is a theorem:

1. $O \to O$;
2. $J \leftrightarrow \big(J \lor (L \land \neg L)\big)$;
3. $((A \to B) \to A) \to A$;
4. $\big((P \to P) \to Q)\big) \to Q$;
5. $\big((C \land D) \leftrightarrow (D \land C)\big)$.

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

E. In this section, I claimed that it is just as hard to show that two sentences are not provably equivalent, as it is to show that a sentence is not a theorem. Why did I claim this? (*Hint*: think of a sentence that would be a theorem iff $\mathcal{A}$ and $\mathcal{B}$ were provably equivalent.)

32

Proof Strategies

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 connective. 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 $\mathcal{A} \to \mathcal{B}$, plan to use the $\to$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 proceeding indirectly If you cannot find a way to show $\mathcal{A}$ directly, try starting by assuming $\neg\mathcal{A}$. If a contradiction follows, then you will be able to obtain $\mathcal{A}$ by $\neg$E. This will often be a good way of proceeding when the conclusion you are aiming at has a *disjunction* as its main connective.

Persist These are guidelines, not laws. Try different things. If one approach fails, then try something else. Remember that if there is one proof, there are many – different proofs that make use of different ideas.

For example: suppose you tried to follow the idea 'work backwards from what you want' in establishing '$P, P \to (P \to Q) \vdash P \to Q$'. You would be tempted to start a subproof from the assumption 'P', and while that proof strategy would eventually succeed, you would have done better to simply apply →E and terminate after one proof step.

By contrast, suppose you tried to follow the idea 'work forwards from what you have' in trying to establish '$(P \vee (Q \vee (R \vee S))) \vdash P \to P$'. You might begin an awkward nested series of subproofs to apply ∨E to the disjunctive premise. But beginning with the conclusion might prompt you instead to simply open a subproof from the assumption P, and the subsequent proof will make no use of the premise at all, as the conclusion is a theorem.

Neither of these heuristics is sacrosanct. You will get a sense of how to construct proofs efficiently and fluently with practice. Unfortunately there is no quick substitute for practice.

Key Ideas in §32

- The nature of natural deduction proofs means that it is sometimes easier to make progress by applying rules to the assumptions, and at other times easier to try and figure out where a conclusion could have come from.
- One may have to try a number of different things in the course of constructing the same proof – there is no simple algorithm to capture logical reasoning in this system.

33

Derived Rules for Sentential

In §§29–28, we introduced the basic rules of our proof system for Sentential. In this section, we shall consider some alternative or additional rules for our system.

None of these rules adds anything fundamentally to our system. They are all DERIVED rules, which means that anything we can prove by using them, we could have proved using just the rules in our original official system of natural deduction proofs. Any of these rules is a *conservative* addition to our proof system, because none of them would enable us to prove anything we could not already prove. (Adding the rules for 'tonk' from §30.2, by contrast, would allow us to prove many new things – any system which includes those rules is not a conservative extension of our original system of proof rules.)

But sometimes adding new rules can shorten proofs, or make them more readable and user-friendly. And some of them are of interest in their own right, as arguably independently plausible rules of implication as they stand, or as alternative rules we could have taken as basic instead.

33.1 Reiteration

The first derived rule is actually one of our main proof rules: *reiteration*. It turns out that we need not have assumed a rule of reiteration. We can replace each application of the reiteration rule on some line $k+1$ (reiterating some prior line m) with the following combination of moves deploying just the *other* basic rules of §§28–29:

m	$\mathcal{A}$	
k	$\mathcal{A} \wedge \mathcal{A}$	$\wedge$I m
$k+1$	$\mathcal{A}$	$\wedge$E k

To be clear: this is not a proof. Rather, it is a proof *scheme*. After all, it uses a variable, $\mathcal{A}$, rather than a sentence of Sentential. But the point is simple. Whatever sentences

of Sentential we plugged in for $\mathcal{A}$, and whatever lines we were working on, we could produce a legitimate proof. So you can think of this as a recipe for producing proofs. Indeed, it is a recipe which shows us that, anything we can prove using the rule R, we can prove (with one more line) using just the *basic* rules of §§29–28. So we can describe the rule R as a derived rule, since its justification is derived from our basic rules.

You might note that in lines 5–7 in the complicated proof in Figure 29.3, we in effect made use of this proof scheme, introducing a conjunction from prior lines only to immediately eliminate again, just to ensure that the relevant sentences appeared directly in the range of the assumption 'Q'.

33.2 Disjunctive Syllogism

Here is a very natural argument form.

> Mitt is either in Massachusetts or in DC. He is not in DC. So, he is in Massachusetts.

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

m	$(\mathcal{A} \vee \mathcal{B})$		m	$(\mathcal{A} \vee \mathcal{B})$	
n	$\neg\mathcal{A}$		n	$\neg\mathcal{B}$	
	$\mathcal{B}$	DS m, n		$\mathcal{A}$	DS m, n

This is, if you like, a new rule of disjunction elimination. But there is nothing fundamentally new here. We can emulate the rule of disjunctive syllogism using our basic proof rules, as this schematic proof indicates:

m	$\mathcal{A} \lor \mathcal{B}$	
n	$\neg\mathcal{A}$	
	$\vdots$	
k	$\quad \mathcal{A}$	
$k+1$	$\quad\quad \neg\mathcal{B}$	
$k+2$	$\quad\quad \mathcal{A}$	R k
$k+3$	$\quad\quad \neg\mathcal{A}$	R n
$k+4$	$\quad \mathcal{B}$	$\neg$E $k+1$–$k+2$, $k+1$–$k+3$
$k+5$	$\quad \mathcal{B}$	
$k+6$	$\quad \mathcal{B}$	R $k+5$
$k+7$	$\mathcal{B}$	$\lor$E m, k–$k+4$, $k+5$–$k+6$

We have used the rule of reiteration in this schematic proof, but we already know that any uses of that rule can themselves be replaced by more roundabout proofs using conjunction introduction and elimination, if required. So adding disjunctive syllogism would not make any new proofs possible that were not already obtainable in our original system.

33.3 *Modus tollens*

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

> If Hillary won the election, then she is in the White House. She is not in the White House. So she did not win the election.

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

This is, if you like, a new rule of conditional elimination.

This rule is, again, a conservative addition to our stock of proof rules. Any application of it could be emulated by this form of proof using our original rules:

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

Again, we made a dispensible use of reiteration.

33.4 Double Negation Elimination

In Sentential, the double negation $\neg\neg\mathcal{A}$ is equivalent to $\mathcal{A}$. In natural languages, too, double negations tend to cancel out – Malcolm is not unaware that his leadership is under threat iff he is aware that it is. That said, you should be aware that context and emphasis can prevent them from doing so. Consider: 'Jane is not *not* happy'. Arguably, one cannot derive '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 Sentential forces us to sacrifice certain nuances of English expressions – we have, in Sentential, just one resource for translating negative expressions like 'not' and the suffix 'un-', even if they are not synonyms in English.

Obviously we can show that $\mathcal{A} \vdash \neg\neg\mathcal{A}$ by means of the following proof:

1	$\mathcal{A}$	
2	$\quad \neg\mathcal{A}$	
3	$\neg\neg\mathcal{A}$	$\neg$I 2–1, 2–2

There is a proof rule that corresponds to the other direction of this equivalence, the rule of DOUBLE NEGATION ELIMINATION:

i	$\neg\neg\mathcal{A}$	
	$\vdots$	
	$\mathcal{A}$	$\neg\neg$E i

This rule is redundant, given the proof rules of Sentential:

1	$\neg\neg\mathcal{A}$	
2	$\quad\neg\mathcal{A}$	
3	$\quad\neg\neg\mathcal{A}$	R 1
4	$\quad\neg\mathcal{A}$	R 2
5	$\mathcal{A}$	$\neg$E 2–4, 2–3

Anything we can prove using the $\neg\neg$E rule can be proved almost as briefly using just $\neg$E.

33.5 *Tertium non datur*

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:

	$\quad\vdots$	
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 roughly '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.

Tertium non datur is able to be emulated using just our original proof rules. Here is a schematic proof which demonstrates this:

i	$\quad\mid\ \mathcal{A}$	
j	$\quad\mid\ \mathcal{B}$	
k	$\quad\mid\ \neg\mathcal{A}$	
l	$\quad\mid\ \mathcal{B}$	
	$\vdots$	
m	$\mathcal{A} \to \mathcal{B}$	$\to$I i–j
$m+1$	$\neg\mathcal{A} \to \mathcal{B}$	$\to$I k–l
$m+2$	$\quad\mid\ \neg\mathcal{B}$	
$m+3$	$\quad\mid\ \quad\mid\ \mathcal{A}$	
$m+4$	$\quad\mid\ \quad\mid\ \mathcal{B}$	$\to$E m, $m+3$
$m+5$	$\quad\mid\ \quad\mid\ \neg\mathcal{B}$	R $m+2$
$m+6$	$\quad\mid\ \neg\mathcal{A}$	$\neg$I $m+3$–$m+5$
$m+7$	$\quad\mid\ \mathcal{B}$	$\to$E $m+1$, $m+6$
$m+8$	$\mathcal{B}$	$\neg$E $m+2$–$m+7$

Again a dispensible use of reiteration occurs in this proof just to make it more readable.

33.6 De Morgan Rules

Our final additional rules are called De Morgan's Laws. (These are named after the nineteenth century logician August De Morgan.) The first two De Morgan rules show the provable equivalence of a negated conjunction and a disjunction of negations.

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

The second pair of De Morgan rules are dual to the first pair: they show the provable equivalence of a negated disjunction and a conjunction of negations.

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

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

The De Morgan rules are no genuine addition to the power of our original natural deduction system. Here is a demonstration of how we could derive the first De Morgan rule:

k	$\neg(\mathcal{A} \land \mathcal{B})$	
m	$\quad \neg(\neg\mathcal{A} \lor \neg\mathcal{B})$	
$m+1$	$\quad\quad \neg\mathcal{A}$	
$m+2$	$\quad\quad \neg\mathcal{A} \lor \neg\mathcal{B}$	$\lor$I $m+1$
$m+3$	$\quad \mathcal{A}$	$\neg$E $m+1$–$m+2$, $m+1$–m
$m+4$	$\quad\quad \neg\mathcal{B}$	
$m+5$	$\quad\quad \neg\mathcal{A} \lor \neg\mathcal{B}$	$\lor$I $m+4$
$m+6$	$\quad \mathcal{B}$	$\neg$E $m+4$–$m+5$, $m+4$–m
$m+7$	$\quad \mathcal{A} \land \mathcal{B}$	$\land$I $m+3$, $m+6$
$m+8$	$\neg\mathcal{A} \lor \neg\mathcal{B}$	$\neg$E m–$m+7$, m–k

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

k	$\neg\mathcal{A} \lor \neg\mathcal{B}$	
m	$\quad \neg\mathcal{A}$	
$m+1$	$\quad\quad \mathcal{A} \land \mathcal{B}$	
$m+2$	$\quad\quad \mathcal{A}$	$\land$E $m+1$
$m+3$	$\quad \neg(\mathcal{A} \land \mathcal{B})$	$\neg$I $m+1$–$m+2$, $m+1$–m
$m+4$	$\quad \neg\mathcal{B}$	
$m+5$	$\quad\quad \mathcal{A} \land \mathcal{B}$	
$m+6$	$\quad\quad \mathcal{B}$	$\land$E $m+5$
$m+7$	$\quad \neg(\mathcal{A} \land \mathcal{B})$	$\neg$I $m+5$–$m+6$, $m+5$–$m+4$
$m+8$	$\neg(\mathcal{A} \land \mathcal{B})$	$\lor$E k, m–$m+3$, $m+4$–$m+7$

Similar demonstrations can be offered explaining how we could derive the third and fourth De Morgan rules. These are left as exercises.

Key Ideas in §33

- Our official system of rules can be augmented by additional rules that are strictly speaking unneccessary – nothing is provable with them that couldn't have been proved without them – but that can nevertheless be used sometimes to speed up proofs.
- Only make use of derived rules when you are told you may do so.
- Some derived rules – such as the rule of double negation elimination – can even be used in place of a rule of our original system, given a different system but with the same things being provable.

Practice exercises

A. The following proofs are missing their commentaries (rule and line numbers). Add them wherever they are required: you may use any of the original or derived rules, as appropriate.

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

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

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

B. Give a proof for each of these arguments:

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

C. Provide proof schemes that justify the addition of the third and fourth De Morgan rules as derived rules.

D. The proofs you offered in response to question **A** above 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.)

34

Alternative Proof Systems for Sentential

We've now developed a system of proof rules, all of which we have supported by showing that they correspond to correct entailments of Sentential. We've also seen that these rules allow us to introduce some derived rules, which make proofs shorter and more convenient but do not allow us to prove anything that we could not have proved already.

This choice of proof system is not forced on us. There are *alternative* proof systems which are nevertheless equivalent to the system we have introduced, in that everything which is provable in our system is provable in the alternative system, and *vice versa*. Indeed, there are lots of alternative systems. In this section, I will discuss just a couple. The alternative systems I will discuss here result from taking one of our derived rules as basic, and showing that doing so allows us to derive a formerly basic rule.

34.1 Replacing Negation Elimination by Double Negation Elimination

The first alternative system results from replacing the rule of negation elimination ¬E by double negation elimination ¬¬E. In combination with the other rules, we can emulate the results of ¬E by an application of ¬I, followed by a single use of ¬¬E:

i	$\quad\vert\ \neg\mathcal{A}$	
	$\quad\vert\ \vdots$	
j	$\quad\vert\ \mathcal{B}$	
	$\quad\vert\ \vdots$	
k	$\quad\vert\ \neg\mathcal{B}$	
$k+1$	$\neg\neg\mathcal{A}$	¬I i–j, i–k
$k+2$	$\mathcal{A}$	¬¬E $k+1$

This proof shows that, in a system with ¬I and ¬¬E, we do not need any separate elimination rule for a single negation – the effect of any such rule could be perfectly simulated by the above schematic proof. The addition of a single negation elimination rule would not allow us to prove any more than we already can. So in a sense, the rules of double negation elimination and negation elimination are equivalent, at least given the other rules in our system.

Since we've shown ¬¬E to be a derived rule in our original system, this alternative system proves exactly the same things as our original system. The proofs will look different, but there is a correct proof of $\mathcal{C}$ from $\mathcal{A}_1, \ldots, \mathcal{A}_n$ in one system, there will be a corresponding proof in the other system. Any proof in the one system can be converted into a proof in the other by replacement of the appropriate instances of the rules.

34.2 Replacing Disjunction Elimination with Disjunctive Syllogism

Another alternative system results from replacing disjunction elimination ∨E (proof by cases) by disjunctive syllogism DS. We already know that DS is a derived rule: any proof which uses disjunctive syllogism can be converted into a proof that uses just the basic rules. So to show the alternative system proves the same things as our original system, we just need to show that we can emulate the effects of ∨E by using DS.

Recall that a schematic proof that makes use of ∨E has the structure in Figure 34.1. To transform this into a proof that makes use instead of DS, we are going to use those two subproofs, but in the scope of an assumption that $\neg\mathcal{C}$. The schematic proof looks like that in Figure 34.2. You can see that the proof is the same as the original at lines i–j and k–l, which mirror the original two subproofs. But they play a quite different role now. The subproof deriving $\mathcal{C}$ from $\mathcal{A}$ is now used in the service of a *reductio* of $\mathcal{A}$, deriving $\neg\mathcal{A}$ on line $j + 1$ in order to put us in a position to apply DS and derive $\mathcal{B}$, and then derive $\mathcal{C}$ in line with the original subproof, which conflicts with the *reductio* assumption $\neg\mathcal{C}$ on line 2, allowing us to use ¬E to derive the same conclusion as in the original proof. This shows us that – at least if we have our negation rules (or their equivalents) – we can replace ∨E with DS and be able to prove all the same things.

There are pros and cons to using DS as our disjunction elimination rule. You can already see that proof by cases is much clunkier – though perhaps you are not too concerned (maybe you suspect proof by cases is not much used in natural argumentation anyway).

Pro Using DS as an elimination rule for ∨ has the nice feature that we eliminate a disjunction in favour of one of its disjuncts, rather than the unprecedented $\mathcal{C}$ that appears as if from nowhere in the original ∨E rule. We also dispense with the use of subproofs in the statement of the disjunction rules.

Con Adopting DS as a basic rule destroys the nice feature of our standard rules that only one connective is used in any rule. DS needs both disjunction and negation. We cannot, therefore, consider a system which lacks negation rules but has disjunction – the rules are no longer *modular*. This is not especially important for

1	$\mathcal{A} \vee \mathcal{B}$	
	$\vdots$	
i	$\quad \mathcal{A}$	
	$\quad \vdots$	
j	$\quad \mathcal{C}$	
	$\vdots$	
k	$\quad \mathcal{B}$	
	$\quad \vdots$	
l	$\quad \mathcal{C}$	
	$\mathcal{C}$	$\vee$E 1, i–j, k–l

Figure 34.1: Schematic $\vee$E proof.

1	$\mathcal{A} \vee \mathcal{B}$	
2	$\quad \neg\mathcal{C}$	
	$\quad \vdots$	
i	$\quad\quad \mathcal{A}$	
	$\quad\quad \vdots$	
j	$\quad\quad \mathcal{C}$	
$j+1$	$\quad\quad \neg\mathcal{C}$	R 2
$j+2$	$\quad \neg\mathcal{A}$	$\neg$I i–j, i–$j+1$
k	$\quad \mathcal{B}$	DS 1, $j+2$
	$\quad \vdots$	
l	$\quad \mathcal{C}$	
$l+1$	$\quad \neg\mathcal{C}$	R 2
	$\mathcal{C}$	$\neg$E 2–l, 2–$l+1$

Figure 34.2: Schematic proof using DS to emulate $\vee$E.

Sentential, but could be important if you go on to consider other logical systems which may vary the rules for one connective independently of all the others. DS shackles negation and disjunction together – even arguments which have, intuitively nothing to do with negation, end up having to use negation in their proof. This is illustrated by the following two proofs that $(A \vee B) \vdash (B \vee A)$, the first using our original system, the second using the system which has DS as a basic rule:

1	$(A \vee B)$	
2	$\quad A$	
3	$\quad (B \vee A)$	$\vee$I 2
4	$\quad B$	
5	$\quad (B \vee A)$	$\vee$I 4
6	$(B \vee A)$	$\vee$E 1, 2–3, 4–5

1	$(A \vee B)$	
2	$\quad \neg(B \vee A)$	
3	$\quad\quad A$	
4	$\quad\quad (B \vee A)$	$\vee$I 3
5	$\quad\quad \neg(B \vee A)$	R 2
6	$\quad \neg A$	$\neg$I 3–4, 3–5
7	$\quad B$	DS 1, 6
8	$\quad (B \vee A)$	$\vee$I 7
9	$\quad \neg(B \vee A)$	R 2
10	$(B \vee A)$	$\neg$E 2–8, 2–9

Chapter 7

Natural Deduction for Quantifier

35

Basic Rules for Quantifier

Quantifier makes use of all of the connectives of Sentential. So proofs in Quantifier will use all of the basic and derived rules from chapter 6. We shall also use the proof-theoretic notions (particularly, the symbol '⊢') introduced in that chapter.

Some arguments in Quantifier don't need any new rules. Consider this:

$$\neg(\forall x Px \lor \exists y Py) \therefore \neg\forall x Px.$$

1	$\neg(\forall x Px \lor \exists y Py)$	
2	$\quad \forall x Px$	
3	$\quad (\forall x Px \lor \exists y Py)$	$\lor$I 2
4	$\quad \neg(\forall x Px \lor \exists y Py)$	R 1
5	$\neg\forall x Px$	$\neg$I 2–3, 2–4

However, not every Quantifier sentence has a Sentential connective as its main connective. So we will also need some new basic rules to govern the quantifiers, and to govern the identity sign, to deal with those sentences where the main connective is a quantifier and where the sentence is an identity predication.

35.1 Universal Elimination

Holding fixed the claim that everything is F, you can conclude that any particular thing is F. *You name it; it's F.* The same is true for many-place predicates: if every human is shorter than 3km tall, then Amy is shorter than 3km tall, and Bob is, and Jonquil is, and everyone else you can name.

Accordingly, the following reasoning should be fine for the corresponding symbolisations in Quantifier:

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

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

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

We obtained line 2 here by dropping the universal quantifier and replacing every instance of 'x' with 'd'. We could have done the same with any other name we wanted.

This motivates the UNIVERSAL ELIMINATION rule ($\forall$E), using the notion for uniform substitution we introduced in §22.4:

m	$\forall x \mathcal{A}$	
	$\vdots$	
	$\mathcal{A}\vert_{c \curvearrowright x}$	$\forall$E m

Where c can be any name.

The intent of the rule is that you can obtain any *substitution instance* of a universally quantified formula: replace every occurrence of the free variable x in $\mathcal{A}$ with any chosen name. (If there are any – the rule is also good when $\mathcal{A}$ has no free variable, because then the quantifier $\forall x$ is redundant.) Remember here that the expression 'c' is a metalanguage variable over names: you are not required to replace the variable x by the Quantifier name 'c', but you can select any name you like!

I should emphasise that (as with every elimination rule) you can only apply the $\forall$E rule when the universal quantifier is the main connective. Thus the following is outright banned:

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

This is illegitimate, since '$\forall x$' is not the main connective in line 1. (If you need a reminder as to why this sort of inference should be banned, reread §16.)

Here is an example of the rule in action. Suppose we wanted to show that $\forall x \forall y(Rxx \to Rxy), Raa \therefore Rab$ is provable. The proof might go like this:

1	$\forall x \forall y(Rxx \to Rxy)$	
2	Raa	
3	$\forall y(Raa \to Ray)$	$\forall$E 1
4	$Raa \to Rab$	$\forall$E 3
5	Rab	$\to$E 4, 2

35.2 Existential Introduction

Given the assumption that some specific named thing is an F, you can conclude that something is an F: 'Sylvester reads, so someone reads' seems like a conclusive argument. So we ought to allow the inference from a claim about some particular thing being F, to a general claim that something or other is F:

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.

There are some pitfalls with this description of what we have done. The following argument is invalid: 'Someone loves Alice; so someone is such that someone loves themselves'. So we ought not to be able to conclude '$\exists x \exists x Rxx$' from '$\exists x Rxa$'. Accordingly, our rule cannot be *replace a name by a variable, and stick a corresponding quantifier out the front* – since that would would permit the proof of the invalid argument.

We take our cue from the $\forall$E rule. This rule says: take a sentence $\forall x \mathcal{A}$, then we can remove the quantifier and substitute an arbitrary name for some free variable in the formula $\mathcal{A}$ (assuming there is one). The $\exists$I rule is in some sense a mirror image of this rule: it allows us to move from a sentence with an arbitrary name – that might be thought of as the result of substituting a name for a free variable in some formula $\mathcal{A}$ – to a quantified sentence $\exists x \mathcal{A}$. So here is how we formulate our rule of EXISTENTIAL INTRODUCTION:

m	$\mathcal{A}\vert_{c\curvearrowright x}$	
	$\vdots$	
	$\exists x\mathcal{A}$	$\exists$I m

So really we should think that the proof just above should be thought of as concluding $\exists xRxxd$ from '$Rxxd$'$\vert_{a\curvearrowright x}$.

If we have this rule, we cannot provide a proof of the invalid argument. For '$\exists xRxa$' is not a substitution instance of '$\exists x\exists xRxx$' – both instances of 'x' in 'Rxx' are bound by the second existential quantifier, so neither is free to be substituted. So the premise is not of the right form for the rule of $\exists$I to apply.

On the other hand, this proof is correct:

1	Raa	
2	$\exists xRax$	$\exists$I 1

Why? Because the assumption 'Raa' is in fact not only a substitution instance of $\exists xRxx$, but also a substitution instance of '$\exists xRax$', since 'Rax'$\vert_{a\curvearrowright x}$ is just 'Raa' too. So we can vindicate the intuitively correct argument 'Narcissus loves himself, so there is someone who loves Narcissus'.

As we just saw, applying this rule requires some skill in being able to recognise substitution instances. Thus the following is allowed:

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

This is okay, because '$Raad$' can arise from substitition of 'a' for 'x' in '$Rxad$', and '$\exists xRxad$' can arise from substitition of 'a' for 'y' in '$\exists xRxyd$'. But this is banned:

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

This is because '$\exists xRxad$' is not a substitution instance of '$\exists x\exists xRxxd$', since (again) both occurrences of 'x' in '$Rxxd$' are already bound and so not available for free substitution.

Here is an example which shows our two proof rules in action, a proof that this argument is correct:

$$\forall x \forall y (Rxy \land Ryx) \therefore \exists x Rxx,$$

1	$\forall x \forall y (Rxy \land Ryx)$	
2	$\forall y (Ray \land Rya)$	$\forall$E 1
3	$(Raa \land Raa)$	$\forall$E 2
4	Raa	$\land$E 3
5	$\exists x Rxx$	$\exists$I 4

For another example, consider this proof of '$\exists x (Px \lor \neg Px)$' from no assumptions:

1	$\quad \neg(Pd \lor \neg Pd)$	
2	$\quad\quad \neg Pd$	
3	$\quad\quad (Pd \lor \neg Pd)$	$\lor$I 2
4	$\quad\quad \neg(Pd \lor \neg Pd)$	R 1
5	$\quad Pd$	$\neg$E 2–3, 2–4
6	$\quad (Pd \lor \neg Pd)$	$\lor$I 5
7	$\quad \neg(Pd \lor \neg Pd)$	R 1
8	$(Pd \lor \neg Pd)$	$\neg$E 1–6, 1–7
9	$\exists x (Px \lor \neg Px)$	$\exists$I 8

35.3 Empty Domains

The following proof combines our two new rules for quantifiers:

1	$\forall x Fx$	
2	Fa	$\forall$E 1
3	$\exists x Fx$	$\exists$I 2

Could this be a bad proof? If anything exists at all, then certainly we can infer that something is F, from the fact that everything is F. But what if *nothing* exists at all? Then it is surely vacuously true that everything is F; however, it does not following that something is F, for there is nothing to *be* F. So if we claim that, as a matter of logic alone, '$\exists x Fx$' follows from '$\forall x Fx$', then we are claiming that, as a matter of *logic alone*, there is something rather than nothing. This might strike us as a bit odd.

Actually, we are already committed to this oddity. In §15, we stipulated that domains in Quantifier must have at least one member. We then defined a logical truth (of Quantifier) as a sentence which is true in every interpretation. Since '$\exists x\, x = x$' will be true in every interpretation, this *also* had the effect of stipulating that it is a matter of logic that there is something rather than nothing.

Since it is far from clear that logic should tell us that there must be something rather than nothing, we might well be cheating a bit here.

If we refuse to cheat, though, then we pay a high cost. Here are three things that we want to hold on to:

- $\forall x Fx \vdash Fa$: after all, that was ∀E.
- $Fa \vdash \exists x Fx$: after all, that was ∃I.
- the ability to copy-and-paste proofs together: after all, reasoning works by putting lots of little steps together into rather big chains.

If we get what we want on all three counts, then we have to countenance that $\forall x Fx \vdash \exists x Fx$. So, if we get what we want on all three counts, the proof system alone tells us that there is something rather than nothing. And if we refuse to accept that, then we have to surrender one of the three things that we want to hold on to!

Before we start thinking about which to surrender,[1] we might want to ask how *much* of a cheat this is. Granted, it may make it harder to engage in theological debates about why there is something rather than nothing. But the rest of the time, we will get along just fine. So maybe we should just regard our proof system (and Quantifier, more generally) as having a very slightly limited purview. If we ever want to allow for the possibility of *nothing*, then we shall have to cast around for a more complicated proof system. But for as long as we are content to ignore that possibility, our proof system is perfectly in order. (As, similarly, is the stipulation that every domain must contain at least one object.)

35.4 Universal Introduction

Suppose you had shown of each particular thing that it is F (and that there are no other things to consider). Then you would be justified in claiming that everything is F. This would motivate the following proof rule. If you had established each and every single substitution instance of '$\forall x Fx$', then you can infer '$\forall x Fx$'.

Unfortunately, that rule would be utterly unusable. To establish each and every single substitution instance would require proving 'Fa', 'Fb', ..., 'Fj_2', ..., 'Fr_{79002}', ..., and so on. Indeed, since there are infinitely many names in Quantifier, this process would never

[1] Many will opt for restricting ∃I. If we permit an empty domain, we will also need 'empty names' – names without a referent. When the name c is empty, it seems problematic to conclude from 'c is F' that there is something which is F. (Does 'Santa Claus drives a flying sleigh' entail 'Someone drives a flying sleigh'?) But empty names are not cost-free; understanding how a name that doesn't name anything can have any meaning at all has vexed many philosophers and linguists.

come to an end. So we could never apply that rule. We need to be a bit more cunning in coming up with our rule for introducing universal quantification.

Our cunning thought will be inspired by considering:

$$\forall x Fx \therefore \forall y Fy$$

This argument should *obviously* be valid. After all, alphabetical variation in choice of variables ought to be a matter of taste, and of no logical consequence. But how might our proof system reflect this? Suppose we begin a proof thus:

1	$\forall x Fx$	
2	Fa	$\forall$E 1

We have proved 'Fa'. And, of course, nothing stops us from using the same justification to prove 'Fb', 'Fc', ..., 'Fj_2', ..., 'Fr_{79002}, ..., and so on until we run out of space, time, or patience. But reflecting on this, we see that this is a way to prove Fc, for any name c. And if we can do it for *any* thing, we should surely be able to say that 'F' is true of *everything*. This therefore justifies us in inferring '$\forall y Fy$', thus:

1	$\forall x Fx$	
2	Fa	$\forall$E 1
3	$\forall y Fy$	$\forall$I 2

The crucial thought here is that 'a' was just some *arbitrary* name. There was nothing special about it – we might have chosen any other name – and still the proof would be fine. And this crucial thought motivates the universal introduction rule ($\forall$I):

m	$\mathcal{A}\vert_{c \rightsquigarrow x}$	
	$\vdots$	
	$\forall x \mathcal{A}$	$\forall$I m

c must not occur in any undischarged assumption, or elsewhere in $\mathcal{A}$

A crucial aspect of this rule, though, is bound up in the accompanying constraint. In English, a name like 'Sylvester' can play two roles: it can be introduced as a name for a specific thing ('let me dub thee Sylvester'!), or as an *arbitrary name*, introduced by this sort of stipulation: *let 'Sylvester' name some arbitrarily chosen man*. The name doesn't tell us, as it subsequently appears, whether it was introduced in one way or the other. But if it was introduced as an arbitrary name, then any conclusions we draw about this Sylvester aren't really dependent on the particular arbitrarily chosen referent – they all

depend rather on the stipulation used in introducing the name, and so (specifically) they will all be consequences of the only fact we know for sure about this Sylvester, that he is male. If all men are mortal, then an arbitrarily chosen man, whom we temporarily call 'Sylvester', is mortal. If Sylvester is mortal, then there is a date he will die. But since he was selected arbitrarily, without reference to any further particulars of his life, then for any man, there exists a date he will die. And that is appropriate reasoning from a universal generalisation, to another generalisation, via claims about a specific but arbitrarily chosen person.[2]

We don't have stipulations to introduce a name as an arbitrary name in Quantifier. But we do have a way of ensuring that the name has no prior associations other than those linked to a prior universal generalisation, if we insist that, when the name is about to be eliminated from the proof, no assumption about what that name denotes is being relied on. That way, we can know that however it was introduced to the proof, it was not done in a way that involved making specific assumptions about whatever the name arbitrarily picks out. The simplest way for this to happen is that the name was introduced by an application of ∀E, as an arbitrary name in the standard sense. But there are other ways too.

This constraint ensures that we are always reasoning at a sufficiently general level. To see the constraint in action, consider this terrible argument:

> Everyone loves Kylie Minogue; therefore everyone loves themselves.

We might symbolise this obviously invalid inference pattern as:

$$\forall x Lxk \therefore \forall x Lxx$$

Now, suppose we tried to offer a proof that vindicates this argument:

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

This is not allowed, because 'k' occurred already in an undischarged assumption, namely, on line 1. The crucial point is that, if we have made any assumptions about the object we are working with (including assumptions embedded in $\mathcal{A}$ itself), then we are not reasoning generally enough to license the use of ∀I.

Although the name may not occur in any *undischarged* assumption, it may occur as a discharged assumption. That is, it may occur in a subproof that we have already closed. For example:

[2] The details about how this sort of arbitrary reference works are interesting. A controversial but nevertheless attractive view of how it might work is Wylie Breckenridge and Ofra Magidor (2012) 'Arbitrary Reference', *Philosophical Studies* **158**, pp. 377–400.

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

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

Here is another proof featuring an application of $\forall$I after discharging an assumption about some name 'a':

1	$\quad Fa \land \neg Fa$	
2	$\quad Fa$	$\land$E 1
3	$\quad \neg Fa$	$\land$E 1
4	$\neg(Fa \land \neg Fa)$	$\neg$I 1–2, 1–3
5	$\forall x \neg(Fx \land \neg Fx)$	$\forall$I 4

Here we were able to derive that something could not be true of a, no matter what a is. We cannot make a coherent assumption that a is both F and isn't F, so it doesn't really matter what 'a' denotes. So the open sentence '$\neg(F \land \neg Fx)$' could not be true of anything at all. That is why we are entitled to discharge that assumption, and then any subsequent use of 'a' in the proof must be depending not on particular facts about this a, but about anything at all, including whatever it is that 'a' happens to pick out.

You might wish to recall the proof of '$\exists x(Px \lor \neg Px)$' from page 307. Note that, by the second-last line, we had already discharged any assumption which relied on the specific name chosen (in that case, 'd'). The existential introduction rule has no constraints on it, so that it was not necessary to discharge any assumptions using the name before applying that rule. But we see now that, since those assumptions were in fact discharged, we could have applied universal introduction at that second last line, to yield a proof of '$\forall x(Px \lor \neg Px)$'.

We can also use our universal rules together to show some things about how quantifier order doesn't matter, when the strings of quantifiers are of the same type. For example

$$\forall x \forall y \forall z Syxz \therefore \forall z \forall y \forall x Syxz$$

can be proved as follows:

1	$\forall x \forall y \forall z Syxz$	
2	$\forall y \forall z Syaz$	$\forall$E 1
3	$\forall z Sbaz$	$\forall$E 2
4	$Sbac$	$\forall$E 3
5	$\forall x Sbxc$	$\forall$I 4
6	$\forall y \forall x Syxc$	$\forall$I 5
7	$\forall z \forall y \forall x Syxz$	$\forall$I 6

Here we successively eliminate the quantifiers in favour of arbitrarily chosen names, and then reintroduce the quantifiers (though in a different order). The only undischarged assumption throughout the proof is the first line, with no names at all, so all of the uses of universal introduction are acceptable.

35.5 Existential Elimination

Suppose we know that *something* is F. The problem is that simply knowing this does not tell us which thing is F. So it would seem that from '$\exists x Fx$' we cannot immediately conclude 'Fa', 'Fe_{23}', or any other substitution instance of the sentence. What can we do?

Suppose we know that something is F, and that everything which is F is G. In (almost) natural English, we might reason thus:

> Since something is F, there is some particular thing which is an F. We do not know anything about it, other than that it's an F, but for convenience, let's call it 'obbie'. So: obbie is F. Since everything which is F is G, it follows that obbie is G. But since obbie is G, it follows that something is G. And nothing depended on which object, exactly, obbie was. So, something is G.

We might try to capture this reasoning pattern in a proof as follows:

1	$\exists x Fx$	
2	$\quad \forall x(Fx \to Gx)$	
3	$\quad\quad Fo$	
4	$\quad\quad Fo \to Go$	$\forall$E 2
5	$\quad\quad Go$	$\to$E 4, 3
6	$\quad\quad \exists x Gx$	$\exists$I 5
7	$\quad \exists x Gx$	$\exists$E 1, 3–6

Breaking this down: we started by writing down our assumptions. At line 3, we made an additional assumption: 'Fo'. This was just a substitution instance of '$\exists xFx$'. On this assumption, we established '$\exists xGx$'. But note that we had made no *special* assumptions about the object named by 'o'; we had *only* assumed that it satisfies 'Fx'. So nothing depends upon which object it is. And line 1 told us that *something* satisfies 'Fx'. So our reasoning pattern was perfectly general. We can discharge the specific assumption 'Fo', and simply infer '$\exists xGx$' on its own.

Putting this together, we obtain the existential elimination rule (∃E):

m	$\exists x\mathcal{A}$	
i	$\quad \mathcal{A}\vert_{c\frown x}$	
	$\quad \vdots$	
j	$\quad \mathcal{B}$	
	$\vdots$	
	$\mathcal{B}$	∃E m, i–j

c must not occur in any assumption undischarged before line i
c must not occur in $\exists x\mathcal{A}$
c must not occur in $\mathcal{B}$

As with universal introduction, the constraints are extremely important. To see why, consider the following terrible argument:

> Tim Button is a lecturer. There is someone who is not a lecturer. So Tim Button is both a lecturer and not a lecturer.

We might symbolise this obviously invalid inference pattern as follows:

$$Lb, \exists x\neg Lx \therefore Lb \land \neg Lb$$

Now, suppose we tried to offer a proof that vindicates this argument:

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

The last line of the proof is not allowed. The name that we used in our substitution instance for '$\exists x \neg Lx$' on line 3, namely 'b', occurs in line 4. And the following proof would be no better:

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

The last line of the proof would still not be allowed. For the name that we used in our substitution instance for '$\exists x \neg Lx$', namely 'b', occurs in an undischarged assumption, namely line 1.

The moral of the story is this.

> If you want to squeeze information out of an existentially quantified claim $\exists x \mathcal{A}$, choose a *new* name, never before used in the proof, to substitute for the variable in $\mathcal{A}$.

That way, you can guarantee that you meet all the constraints on the rule for $\exists$E.

An argument that makes use of both patterns of arbitrary reasoning is this example due to Breckenridge and Magidor: 'from the premise that there is someone who loves everyone to the conclusion that everyone is such that someone loves them'. Here is a proof in our system, letting 'Lxy' symbolise '$\underset{1}{\underline{\quad}}$ loves $\underset{2}{\underline{\quad}}$', letting '$h$' symbolise the arbitrarily chosen name 'Hiccup' and letting 'a' symbolise the arbitrarily chosen name 'Astrid':

1	$\exists x \forall y Lxy$	
2	$\quad \forall y Lhy$	
3	$\quad Lha$	$\forall$E 2
4	$\quad \exists x Lxa$	$\exists$I 3
5	$\quad \forall y \exists x Lxy$	$\forall$I 4
6	$\forall y \exists x Lxy$	$\exists$E 1, 2–5

At line 3, both our arbitrary names are in play – h was newly introduced to the proof in line 2 as the arbitrary person Hiccup who witnesses the truth of '$\exists x \forall y Lxy$', and 'a'

at line 3 as an arbitrary person Astrid beloved by Hiccup. We can apply ∃I without restriction at line 4, which takes the name 'h' out of the picture – we no longer rely on the specific instance chosen, since we are back at generalities about someone who loves everyone, being such that they also love the arbitrarily chosen someone Astrid. So we can safely apply ∀I at line 5, since the name 'a' appears in no assumption nor in '$\forall y \exists x Lxy$'. But now we have at line 5 a claim that doesn't involve the arbitrary name 'h' either, which was newly chosen to not be in any undischarged assumption or in $\exists x \forall y Lxy$. So we can safely say that the name Hiccup was just arbitrary, and nothing in the proof of '$\forall y \exists x Lxy$' depended on it, so we can discharge the specific assumption about h that was used in the course of that proof and nevertheless retain our entitled ment to '$\forall y \exists x Lxy$'.

35.6 The Barber

Let's try something a little more complicated, the so-called 'Barber paradox':

> in a certain remote Sicilian village, approached by a long ascent up a precipitous mountain road, the barber shaves all and only those villagers who do not shave themselves. Who shaves the barber? If he himself does, then he does not (since he shaves only those who do not shave themselves); if he does not, then he indeed does (since he shaves all those who do not shave themselves). The unacceptable supposition is that there is such a barber – one who shaves himself if and only if he does not. The story may have sounded acceptable: it turned our minds, agreeably enough, to the mountains of inland Sicily. However, once we see what the consequences are, we realize that the story cannot be true: there cannot be such a barber, or such a village. The story is unacceptable.[3]

This uses some of our tricky quantifer rules, disjunction elimination (proof by cases) and negation introduction (*reductio*), so it is really a showcase of many things we've learned so far.

Let's first try to symbolise the argument.

Domain: residents of a certain remote Sicilian village
B: $\underbrace{____}_{1}$ is a barber
S: $\underbrace{____}_{1}$ shaves $\underbrace{____}_{2}$

The argument then revolves around the claim that there is a barber who shaves everyone who doesn't shave themselves. Semi-formally paraphrased: someone x exists such that x is a barber and for all people y: y does not shave themselves iff x shaves y. That is:

$$\exists x(Bx \land \forall y(\neg Syy \leftrightarrow Sxy)).$$

The argument takes the form of a *reductio*, so we will begin the proof by assuming this claim for the sake of argument and see what happens:

3 R M Sainsbury (2009) *Paradoxes* 3rd ed., Cambridge University Press, pages 1–2.

1	$\exists x(Bx \land \forall y(\neg Syy \leftrightarrow Sxy))$	
2	$(Ba \land \forall y(\neg Syy \leftrightarrow Say))$	
3	$\forall y(\neg Syy \leftrightarrow Sxy)$	∧E 2
4	$(\neg Saa \leftrightarrow Saa)$	∀E 3
5	$\neg(P \land \neg P)$	
6	$\neg Saa$	
7	Saa	↔E 4, 6
8	$\neg Saa$	R 6
9	Saa	¬E 6–7, 6–8
10	$\neg Saa$	↔E 4, 9
11	$(P \land \neg P)$	¬E 5–9, 5–10
12	$(P \land \neg P)$	∃E 1, 2–11
13	P	∧E 12
14	$\neg P$	∧E 12
15	$\neg\exists x(Bx \land \forall y(\neg Syy \leftrightarrow Sxy))$	¬I 1–13, 1–14

One trick to this proof is to be sure to instantiate the universally quantified claim at line 3 by using the same name 'a' as was already used in line 2. This is because, intuitively, the problem case for this supposed barber arises when you think about whether they shaves themselves or not. But themselves trickiest part of this proof occurs at lines 5–11. By line 4, we've already derived a contradictory biconditional. But if we just use it to derive 'Saa' and '$\neg Saa$', the contradictory claims we obtain would end up involving the name 'a'. That would mean we couldn't apply the ∃E rule, since the final line of the subproof would contain the chosen name, so we couldn't get our logical falsehood out of the subproof beginning on line 2, and hence could perform the desired *reductio* on line 1 via ¬I. So our trick is to suppose the negation of an *unrelated* logical falsehood on line 5, derive the logical falsehood from line 4 in the range of that assumption, and hence use ¬E to derive the logical falsehood '$P \land \neg P$' on line 11. This doesn't contain the name 'a', and hence can be extracted from the subproof to show that line 1 by itself suffices to derive a logical falsehood, and that shows the supposition that there is such a barber is a logical falsehood.

35.7 Justification of these Quantifier Rules

Above, I offered informal arguments for each of our quantifier rules that seem to exemplify the pattern of argument in the rule, and to be intuitively valid. But we can also offer justifications for our rules in terms of interpretations of the sentences involved, and the principles governing truth of quantified sentences introduced in §22.4.

For example, consider any interpretation which makes $\forall x\mathcal{A}$ true. In any such interpretation, there will be a nonempty domain, and every name will denote some member of this domain. $\forall x\mathcal{A}$ is true just in case for any name we like, it will denote something of which $\mathcal{A}|_{c\frown x}$ is true. So in any such interpretation, for each name in the language c, $\mathcal{A}|_{c\frown x}$ will also be true. So the proof rule of ∀E corresponds to a valid argument form.

For ∃E, the case is only a little more involved. Suppose $\exists x\mathcal{A}$ is true in an interpretation. Then there is some interpretation, otherwise just like the original one, in which some new name c is assigned to some object in the domain, and where $\mathcal{A}|_{c\frown x}$ is true. Suppose that, in fact, every interpretation which makes $\mathcal{A}|_{c\frown x}$ true also makes $\mathcal{B}$ true, where the new name c does not appear in $\mathcal{B}$. Could $\mathcal{B}$ be false in our original interpretation? No – for everything that appears in $\mathcal{B}$ is already interpreted in the original interpretation, with the same interpretation as in the interpretation which makes it true. So it must be true in our original interpretation too. So $\mathcal{C}_1, ..., \mathcal{C}_n, \exists x\mathcal{A} \vDash \mathcal{B}$ (when the name c makes no appearance in any sentence in this argument), and the proof rule of ∃E corresponds to a valid argument form.

You may also offer arguments from intepretations to the effect that our other quantifier proof rules correspond to valid arguments in Quantifier:

- $\mathcal{A}|_{c\frown x} \vDash \exists x\mathcal{A}$ – if c is used in the proof, it must have an interpretation as something in the domain, and so something in the domain satisfies $\mathcal{A}$;
- If this entailment holds:
$$\mathcal{C}_1, ..., \mathcal{C}_n \vDash \mathcal{A}|_{c\frown x},$$
where the name c occurs nowhere among $\mathcal{C}_i$ or elsewhere in $\mathcal{A}$, then this entailment also holds:
$$\mathcal{C}_1, ..., \mathcal{C}_n \vDash \forall x\mathcal{A}.$$
For we could have substituted any other name for c and the original entailment would still have succeeded, since it could not have depended on the specific name chosen. So it doesn't matter what the interpretation of c happens to be, and if that doesn't matter, it must be because everything is $\mathcal{A}$.

So we are again comforted: our proof rules can never lead us from true assumptions to false claims, if correctly applied.

35.8 Proof-Theoretic Concepts in Quantifier

The notion of provability of $\mathcal{A}$ from undischarged assumptions $\mathcal{C}_1, ..., \mathcal{C}_n$ was introduced for Sentential in §31. The rules for Quantifier are different, but the earlier definitions go through unchanged, once we remember that the notion of proof in Quantifier means: *provable using the rules for Sentential and those for Quantifier*.

So in what follows I will make use of the single turnstile '⊢' to mean that there is a correctly formed proof using only the rules of Sentential and Quantifier. (And once we introduce the identity rules in §37, I will tacitly assume that proofs can make use of those rules too in justifying a claim using '⊢'.)

Likewise, the notions of theoremhood, provable equivalence, and joint contrariness all carry over their definitions, because they are defined in terms of the single turnstile.

Key Ideas in §35

- We augment our natural deduction proof system for Sentential by adding rules governing quantifiers to go partway towards a natural deduction system for Quantifier.
- The rules for ∀E and ∃I are straightforward and can be applied regardless of which names we deploy.
- But the other quantifier rules ∀I and ∃E contain some important restrictions on which names we can use. These restrictions are motivated by considerations about arbitrary reference which inform us when we can introduce 'dummy names' in the course of our proofs and what we can do with them.
- Our proof rules match the interpretation of Quantifier we have given – they will not permit us to say that some claim is provable from some assumptions when that claim isn't entailed by those assumptions.
- The notation '⊢' for provability carries over from its earlier use in Sentential unchanged, once we understand that a proof can now use the new rules for the new logical connectives in Quantifier.

Practice exercises

A. The following three 'proofs' are *incorrect*. Explain why they are incorrect. If the argument 'proved' is invalid, provide an interpretation which shows that the assumptions involved do not entail the conclusion:

1.

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

2.

1	$\forall x \exists y Rxy$	
2	$\exists y Ray$	∀E 1
3	$\quad Raa$	
4	$\quad \exists x Rxx$	∃I 3
5	$\exists x Rxx$	∃E 2, 3–4

	1	$\exists y \neg(Ty \vee \neg Ty)$	
	2	$\quad \neg(Td \vee \neg Td)$	
	3	$\quad\quad Td$	
	4	$\quad\quad Td \vee \neg Td$	$\vee$I 3
	5	$\quad\quad \neg(Td \vee \neg Td)$	R 2
3.	6	$\quad \neg Td$	$\neg$E 3–4, 3–5
	7	$\quad (Td \vee \neg Td)$	$\vee$I 6
	8	$\quad \neg(Td \vee \neg Td)$	R 2
	9	$\quad ((Td \vee \neg Td) \wedge \neg(Td \vee \neg Td))$	$\wedge$I 7, 8
	10	$((Td \vee \neg Td) \wedge \neg(Td \vee \neg Td))$	$\exists$E 1, 2–9
	11	$\neg\exists y \neg(Ty \vee \neg Ty)$	$\neg$I 1–10

B. The following three proofs are missing their commentaries (rule and line numbers). Add them, to turn them into bona fide proofs.

1	$\forall x \exists y(Rxy \lor Ryx)$
2	$\quad \forall x \neg Rmx$
3	$\quad \exists y(Rmy \lor Rym)$
4	$\quad\quad Rma \lor Ram$
5	$\quad\quad\quad Ram$
6	$\quad\quad\quad Rma$
7	$\quad\quad\quad\quad \neg Ram$
8	$\quad\quad\quad\quad \neg Rma$
9	$\quad\quad\quad Ram$
10	$\quad\quad Ram$
11	$\quad\quad \exists x Rxm$
12	$\quad \exists x Rxm$

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

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

C. In §16 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 Quantifier and offer proofs.

- **Barbari.** Something is H. All G are F. All H are G. So: Some H is F
- **Celaront.** Something is H. No G are F. All H are G. So: Some H is not F
- **Cesaro.** Something is H. No F are G. All H are G. So: Some H is not F.
- **Camestros.** Something is H. All F are G. No H are G. So: Some H is not F.
- **Felapton.** Something is G. No G are F. All G are H. So: Some H is not F.
- **Darapti.** Something is G. All G are F. All G are H. So: Some H is F.
- **Calemos.** Something is H. All F are G. No G are H. So: Some H is not F.
- **Fesapo.** Something is G. No F is G. All G are H. So: Some H is not F.
- **Bamalip.** Something is F. All F are G. All G are H. So: Some H are F.

E. Provide a proof of each claim.

1. $\vdash \forall x Fx \lor \neg \forall x Fx$
2. $\vdash \forall z(Pz \lor \neg Pz)$
3. $\forall x(Ax \to Bx), \exists x Ax \vdash \exists x Bx$
4. $\forall x(Mx \leftrightarrow Nx), Ma \land \exists x Rxa \vdash \exists x Nx$
5. $\forall x \forall y Gxy \vdash \exists x Gxx$
6. $\vdash \forall x Rxx \to \exists x \exists y Rxy$
7. $\vdash \forall y \exists x(Qy \to Qx)$
8. $Na \to \forall x(Mx \leftrightarrow Ma), Ma, \neg Mb \vdash \neg Na$
9. $\forall x \forall y(Gxy \to Gyx) \vdash \forall x \forall y(Gxy \leftrightarrow Gyx)$
10. $\forall x(\neg Mx \lor Ljx), \forall x(Bx \to Ljx), \forall x(Mx \lor Bx) \vdash \forall x Ljx$

F. Write a 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 logically equivalent.

1. $\forall x Px \to Qc, \forall x(Px \to Qc)$
2. $\forall x \forall y \forall z Bxyz, \forall x Bxxx$
3. $\forall x \forall y Dxy, \forall y \forall x Dxy$
4. $\exists x \forall y Dxy, \forall y \exists x Dxy$
5. $\forall x(Rca \leftrightarrow Rxa), Rca \leftrightarrow \forall x Rxa$

H. For each of the following arguments: If it is valid in Quantifier, give a proof. If it is invalid, construct an interpretation to show that it is invalid.

1. $\exists y \forall x Rxy \therefore \forall x \exists y Rxy$
2. $\exists x(Px \wedge \neg Qx) \therefore \forall x(Px \rightarrow \neg Qx)$
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. $\forall x Px \rightarrow \forall x Qx, \exists x \neg Px \therefore \exists x \neg Qx$

36

Derived Rules for Quantifier

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. But they are no substantive addition to our basic rules: for each of the proposed additions, it can be shown that their role in any proof can be wholly emulated by some suitable applications of our basic rules from §35. (The point here is as in §33.)

36.1 Conversion of Quantifiers

In §15, 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 two rules, one for each direction of the equivalence:

m	$\forall x\neg\mathcal{A}$		m	$\neg\exists x\mathcal{A}$	
	$\neg\exists x\mathcal{A}$	$\mathrm{CQ}_{\forall/\neg\exists}\ m$		$\forall x\neg\mathcal{A}$	$\mathrm{CQ}_{\neg\exists/\forall}\ m$

Here is a schematic proof corresponding to our first conversion of quantifiers rule, $\mathrm{CQ}_{\forall/\neg\exists}$:

1	$\forall x \neg \mathcal{A}$	
2	$\quad \exists x \mathcal{A}$	
3	$\quad\quad \mathcal{A}\vert_{c \curvearrowright x}$	
4	$\quad\quad\quad \neg(\mathcal{B} \wedge \neg\mathcal{B})$	
5	$\quad\quad\quad \neg\mathcal{A}\vert_{c \curvearrowright x}$	$\forall$E 1
6	$\quad\quad \mathcal{B} \wedge \neg\mathcal{B}$	$\neg$I 4–5, 4–3
7	$\quad \mathcal{B} \wedge \neg\mathcal{B}$	$\exists$E 2, 3–6
8	$\neg\exists x \mathcal{A}$	$\neg$I 2–7

A couple of things to note about this proof.

1. I was hasty at line 9 – officially I ought to have applied $\wedge$E to line 8, obtaining the contradictory conjuncts in the subproof, and then applied $\neg$I to the assumption opening that subproof. (But then the proof would have gone over the page.)

2. Note that we had to introduce the new name c at line 3. Once we did so, there was no obstacle to applying $\forall$E on that newly introduced name in line 5. But if we had done things the other way around, applying $\forall$E first to some new name c, we would have had to open the subproof with yet another new name d.

3. The sentence $\mathcal{B}$ cannot contain the name c if the application of $\exists$E at line 8 is to be correct. We introduce this arbitrary logical falsehood precisely so we can show that the contradictoriness of our initial assumptions does not depend on the particular choice of name. The alternative would have been to show that the assumption $\mathcal{A}\vert_{c \curvearrowright x}$ leads to logical falsehood, and then applied $\neg$I – but that would have left the name c outside the scope of a subproof and would not have allowed us to apply $\exists$E.

A similar schematic proof could be offered for the second conversion rule, $\text{CQ}_{\neg\exists/\forall}$.

Equally, we might add rules corresponding to the equivalence of $\exists x \neg \mathcal{A}$ and $\neg\forall x \mathcal{A}$:

m	$\exists x \neg \mathcal{A}$		m	$\neg\forall x \mathcal{A}$	
	$\neg\forall x \mathcal{A}$	$\text{CQ}_{\exists/\neg\forall}\ m$		$\exists x \neg \mathcal{A}$	$\text{CQ}_{\neg\forall/\exists}\ m$

Here is a schematic basic proof showing that the third conversion of quantifiers rule just introduced, $\text{CQ}_{\exists/\neg\forall}$, can be emulated just using the standard quantifier rules in combination with the other rules of our system, in which some of the same issues arise as in the earlier schematic proof:

1	$\exists x \neg \mathcal{A}$	
2	$\quad \forall x \mathcal{A}$	
3	$\quad\quad \neg \mathcal{A}\vert_{c \curvearrowright x}$	
4	$\quad\quad\quad \neg(\mathcal{B} \wedge \neg \mathcal{B})$	
5	$\quad\quad\quad \mathcal{A}\vert_{c \curvearrowright x}$	$\forall$E 2
6	$\quad\quad\quad \neg \mathcal{A}\vert_{c \curvearrowright x}$	R 3
7	$\quad\quad \mathcal{B} \wedge \neg \mathcal{B}$	$\neg$E 4–5, 4–6
8	$\quad \mathcal{B} \wedge \neg \mathcal{B}$	$\exists$E 1, 3–7
9	$\neg \forall x \mathcal{A}$	$\neg$I 2–8

A similar schematic proof can be offered for the final CQ rule.

36.2 Alternative Proof Systems for Quantifier

We saw in §34 that it is possible to formulate alternative proof systems that can nevertheless establish the same arguments are provable in Sentential. The same is true for Quantifier. The idea is to get rid of the rules for one quantifier, retaining the rules governing the other quantifier, but then to take the conversion of quantifier rules as basic.

So, for example, we could consider the system which has $\exists$I and $\exists$E, and also has $\text{CQ}_{\exists/\neg\forall}$ and $\text{CQ}_{\neg\forall/\exists}$. With these rules, we can emulate $\forall$E and $\forall$I. A schematic proof showing how to emulate $\forall$E using our other basic rules is this:

1	$\forall x \mathcal{A}$	
2	$\quad \neg \mathcal{A}\vert_{c \curvearrowright x}$	
3	$\quad \exists x \neg \mathcal{A}$	$\exists$I 2
4	$\quad \neg \forall x \mathcal{A}$	$\text{CQ}_{\exists/\neg\forall}$ 3
5	$\quad \forall x \mathcal{A}$	R 1
6	$\mathcal{A}\vert_{c \curvearrowright x}$	$\neg$E 2–5, 2–4

A schematic proof emulating $\forall$I using our other basic rules is trickier. Here it is:

1	Γ	
	$\vdots$	
m	$\quad \neg\forall x\mathcal{A}$	
$m+1$	$\quad \exists x\neg\mathcal{A}$	$\text{CQ}_{\neg\forall/\exists}\ m$
$m+2$	$\quad\quad \neg\mathcal{A}\vert_{c\curvearrowright x}$	
$m+3$	$\quad\quad\quad \neg(\mathcal{B}\wedge\neg\mathcal{B})$	
	$\quad\quad\quad \vdots$	
n	$\quad\quad\quad \mathcal{A}\vert_{c\curvearrowright x}$	Original proof from 1
$n+1$	$\quad\quad\quad \neg\mathcal{A}\vert_{c\curvearrowright x}$	R $m+2$
$n+2$	$\quad\quad \mathcal{B}\wedge\neg\mathcal{B}$	$\neg$E $m+3$–n, $m+3$–$n+1$
$n+3$	$\quad \mathcal{B}\wedge\neg\mathcal{B}$	$\exists$E $m+1$, $m+2$–$n+2$
$n+4$	$\forall x\mathcal{A}$	$\neg$E m–$n+3$

To understand this schematic proof, what we need to remember is that, in order for the original $\forall$I rule to apply, we must already have a proof of $\mathcal{A}\vert_{c\curvearrowright x}$ which relies on assumptions Γ that do not mention c at all. The trick is to make use of that proof *inside* an assumption about an existential witness. We don't try to perform that proof to derive $\mathcal{A}\vert_{c\curvearrowright x}$ and then attempt to manipulate $\neg\forall x\mathcal{A}$ to generate a logical falsehood. Rather, we first assume $\neg\forall x\mathcal{A}$, apply quantifier conversion to obtain $\exists x\neg\mathcal{A}$, assume that c witnesses that existential claim so that $\neg\mathcal{A}\vert_{c\curvearrowright x}$, and then use our original proof to derive $\mathcal{A}\vert_{c\curvearrowright x}$ at line n. To avoid problems with the name appearing at the bottom of the existential witness subproof, we perform the same trick of assuming the falsehood of an arbitrary logical falsehood (so long as $\mathcal{B}$ doesn't include c), and then we manage to derive from Γ what we had hoped to: that $\forall x\mathcal{A}$.

Key Ideas in §36

- The derived rules for Quantifier concern the interaction of quantifiers with negation.
- Do not make use of these derived rules unless you are explicitly told you may do so.

Practice exercises

A. Show that the following are jointly contrary:

1. $Sa \to Tm, Tm \to Sa, Tm \wedge \neg Sa$
2. $\neg\exists x Rxa, \forall x\forall y Ryx$
3. $\neg\exists x\exists y Lxy, Laa$
4. $\forall x(Px \to Qx), \forall z(Pz \to Rz), \forall y Py, \neg Qa \wedge \neg Rb$

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

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

C. In §16, I considered what happens when we move quantifiers 'across' various connectives. Show that each pair of sentences is provably equivalent:

1. $\forall x(Fx \land Ga), \forall x Fx \land Ga$
2. $\exists x(Fx \lor Ga), \exists x Fx \lor Ga$
3. $\forall x(Ga \to Fx), Ga \to \forall x Fx$
4. $\forall x(Fx \to Ga), \exists x Fx \to Ga$
5. $\exists x(Ga \to Fx), Ga \to \exists x Fx$
6. $\exists x(Fx \to Ga), \forall x Fx \to Ga$

NB: the variable 'x' does not occur in 'Ga'.

When all the quantifiers occur at the beginning of a sentence, that sentence is said to be in *prenex normal form*. These equivalences are sometimes called *prenexing rules*, since they give us a means for putting any sentence into prenex normal form.

D. Offer proofs which justify the addition of the other CQ rules as derived rules.

37

Rules for Identity

37.1 Identity Introduction

In §21, I mentioned the philosophically contentious thesis of the *identity of indiscernibles*. This is the claim that objects which are indiscernible in every way – which means, for us, that exactly the same predicates are true of both objects – are, in fact, identical to each other. I also mentioned that in Quantifier, this thesis is not true. There are interpretations in which, for each property denoted by any predicate of the language, two distinct objects either both have that property, or both lack that property. They may well differ on some property 'in reality', but there is nothing in Quantifier which guarantees that that property is assigned as the interpretation of any predicate of the language. It follows that, no matter how many sentences of Quantifier about those two objects I assume, those sentences will not entail that these distinct objects are identical (luckily for us). Unless, of course, you tell me that the two objects are, in fact, identical. But then the argument will hardly be very illuminating.

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

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

$c = c$	=I

Notice that this rule does not require referring to any prior lines of the proof, nor does it rely on any assumptions. For any name c, you can write $c = c$ at any point, with only the =I rule as justification.

Recall that a relation is reflexive iff it holds between anything in the domain and itself (§§18.5 and21.10). Let's see this rule in action, in a proof that identity is reflexive:

1	$a = a$	=I
2	$\forall x x = x$	∀I 1

This seems like magic! But note that the first line is not an assumption (there is no horizontal line), and hence not an undischarged assumption. So the constant 'a' appears in no undischarged assumption or anywhere in the proof other than in '$x = x$'$|_{a \curvearrowright x}$, so the conclusion '$\forall x x = x$' follows by legitimate application of the ∀I rule.

37.2 Identity Elimination

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

m	$a = b$	
n	$\mathcal{A}\|_{a \curvearrowright x}$	
	$\mathcal{A}\|_{b \curvearrowright x}$	=E m, n

This uses our standard notion of substitution – it basically says that if you have some sentence which arises from substituting a for some variable in a formula, then you are entitled to another substitution instance of the same formula using b instead. 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.

Note that nothing in the rule forbids the constant b from occurring in $\mathcal{A}$. So this is a perfectly good instance of the rule:

1	$a = b$	
2	Rab	
3	Rbb	=E 1, 2

Here, 'Rab' is 'Rxb'$|_{a \curvearrowright x}$, and the conclusion 'Rbb' is 'Rxb'$|_{b \curvearrowright x}$, which conforms to the rule. This formulation allows us, in effect, to replace some-but-not-all occurrences of a

name in a sentence by a co-referring name. This rule is sometimes called *Leibniz's Law* – though recall §18.5, where we used that name for a claim about the interpretation of '='.

To see the rules in action, we shall prove some quick results. Recall that a relation is symmetric iff whenever it holds between x and y in one direction, it holds also between y and x in the other direction (§21.9). This condition can be expressed as a sentence of Quantifier:

So first, we shall prove that identity is symmetric, a result we already noted on semantic grounds in §18.5:

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

Line 2 is just '$x = a$'$|_{a \curvearrowright x}$, as well as being of the right form for =I, and line 3 is just $x = a$'$|_{b \curvearrowright x}$, so the move from 2 to 3 is in conformity with the =E rule given the opening assumption.

Having noted the symmetry of identity, note that we can use this to establish the following schematic proof that allows us to use $a = b$ to also move from a claim about b to a claim about a, not just *vice versa* as in our =E rule.:

m	$a = b$	
$m + 1$	$a = a$	=I
$m + 2$	$b = a$	=E m, $m + 1$
n	$\mathcal{A}\|_{b \curvearrowright x}$	
	$\mathcal{A}\|_{a \curvearrowright x}$	=E $m + 2$, n

This schematic proof actually justifies this derived rule, to save time:

m	$a = b$	
n	$\mathcal{A}\|_{b \curvearrowright x}$	
	$\mathcal{A}\|_{a \curvearrowright x}$	=ES m, n

A relation is TRANSITIVE iff whenever it holds between x and y and between y and z, it also holds between x and z. (In the directed graph representation of the relation introduced in §21.9, if there is a path along arrows going from node a to node b via a third node, there is also a direct arrow from a to b.) Second, we shall prove that identity is *transitive*:

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

We obtain line 4 by replacing 'b' in line 3 with 'c'; this is justified given line 2, '$b = c$'. We could alternatively have used the derived rule =ES to replace 'b' in line 2 with 'a', justified by line 3, '$a = b$'.

A relation that is reflexive, symmetric, and transitive is known as an EQUIVALENCE RELATION. So we've proved that identity is an equivalence relation.

Definite Descriptions in Proofs

I want to close this section by giving an example of how proofs involving definite descriptions work. Remember from §19 that we symbolised a definite description 'the F is G', following Russell, as

$$\exists x(\mathcal{F}x \land \forall y(\mathcal{F}y \leftrightarrow x = y) \land \mathcal{G}x).$$

(I omit interior parentheses to aid legibility.)

Using this kind of approach, let us consider this argument:

> The F is G; the H isn't G; so the F isn't H.

The argument, symbolised, looks like this:

$$\exists x(Fx \land \forall y(Fy \leftrightarrow x = y) \land Gx), \exists x(Hx \land \forall y(Hy \leftrightarrow x = y) \land \neg Gx) \vdash \exists x(Fx \land \forall y(Fy \leftrightarrow x = y) \land \neg Hx).$$

Here's the proof:

1	$\exists x(Fx \land \forall y(Fy \leftrightarrow x = y) \land Gx)$	
2	$\exists x(Hx \land \forall y(Hy \leftrightarrow x = y) \land \neg Gx)$	
3	$(Fa \land \forall y(Fy \leftrightarrow a = y) \land Ga)$	
4	$(Hb \land \forall y(Hy \leftrightarrow b = y) \land \neg Gb)$	
5	Ha	
6	$\forall y(Hy \leftrightarrow b = y)$	$\land$E 4
7	$(Ha \leftrightarrow b = a)$	$\forall$E 6
8	$b = a$	$\leftrightarrow$E 7, 5
9	Ga	$\land$E 3
10	$\neg Gb$	$\land$E 4
11	$\neg Ga$	$=$E 8, 10
12	$\neg Ha$	$\neg$I 5–9, 5–11
13	$\neg Ha$	$\exists$E 2, 4–12
14	$(Fa \land \forall y(Fy \leftrightarrow a = y))$	$\land$E 3
15	$(Fa \land \forall y(Fy \leftrightarrow a = y) \land \neg Ha)$	$\land$I 14, 13
16	$\exists x(Fx \land \forall y(Fy \leftrightarrow x = y) \land \neg Hx)$	$\exists$I 15
17	$\exists x(Fx \land \forall y(Fy \leftrightarrow x = y) \land \neg Hx)$	$\exists$E 1, 3–16

Key Ideas in §37

- Adding rules for identity completes our presentation of the natural deduction system for Quantifier.
- The identity elimination rule is basically Leibniz' Law. The identity introduction rule expresses how trivial identity is when the same name flanks the identity symbol.
- We can prove all the properties we traditionally ascribe to identity – reflexivity, symmetry, and transitivity – so identity is an equivalence relation.

Practice exercises

A. Provide a proof of each of the following.

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. Show that the following are provably equivalent:

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

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

C. In §18, I claimed 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.)

D. 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.

E. What condition on the directed graph of a relation corresponds to that relation being an equivalence relation?

38

Proof-Theoretic Concepts and Semantic Concepts

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 $\mathcal{C}$ (and no undischarged assumptions other than $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$). 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 (or interpretation) 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*.

I cannot emphasise enough that these are different notions. But I can emphasise it a bit more: *They are different notions.*

Once you have internalised this point, continue reading.

38.1 Connecting Entailment and Provability

Although our semantic and proof-theoretic notions are different, there is a deep connection between them. To explain this connection, I shall start by considering the relationship between logical truths and theorems.

To show that a sentence is a theorem, you need only perform a proof. Granted, it may be hard to produce a twenty line proof, but it is not so hard to check each line of the proof and confirm that it is legitimate; and if each line of the proof individually is legitimate, then the whole proof is legitimate. Showing that a sentence is a logical truth, though, requires reasoning about all possible interpretations. Given a choice

between showing that a sentence is a theorem and showing that it is a logical truth, it would be easier to show that it is a theorem.

Contrawise, to show that a sentence is *not* a theorem is hard. We would need to reason about all (possible) proofs. That is very difficult. But to show that a sentence is not a logical truth, you need only construct an interpretation in which the sentence is false. Granted, it may be hard to come up with the interpretation; but once you have done so, it is relatively straightforward to check what truth value it assigns to a sentence. Given a choice between showing that a sentence is not a theorem and showing that it is not a logical truth, it would be easier to show that it is not a logical truth.

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

More generally, we have the following powerful result:

$$\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{B} \textbf{ iff } \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{B}.$$

This shows that, whilst provability and entailment are *different* notions, they are extensionally equivalent, holding between just the same sentences in our languages. As such:

- An argument is *valid* iff *the conclusion can be proved from the premises*.
- Two sentences are *logically equivalent* iff they are *provably equivalent*.
- Sentences are *jointly consistent* 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 – let me repeat this – 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.

We showed part of this result along the way, actually. All those little observations I made about how our proof rules were good each took the form of an argument that whenever $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vdash \mathcal{B}$, then also $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n \vDash \mathcal{B}$. This is sometimes known as the SOUNDNESS of our proof system, that whenever you can prove something, there is a corresponding valid argument.

38.2 Next Steps in Logic

The other half of this result is more demanding. This is the COMPLETENESS of our proof system, that whenever there is a valid argument, there is a proof. It is somewhere around this point, of demonstrating the completeness half of the agreement between provability and semantic entailment, when introductory logic becomes intermediate logic. More than one of the authors of this book have written distinct sequels in which the completeness of Sentential is established, along with other results of a 'metalogical' flavour. The interested student is directed to these texts:

- Tim Button's book is *Metatheory*; it covers just Sentential, and some of the results there were actually covered in this book already, notably results around expressiveness of Sentential in §14. *Metatheory* is available at `www.homepages.ucl.ac.uk/~uctytbu/Metatheory.pdf` It should be accessible for self-study by students who have successfully completed this course.
- Antony Eagle's book *Elements of Deductive Logic* goes rather further than *Metatheory*, including direct proofs of Compactness for Sentential, consideration of alternative derivation systems, discussion of computation and decidability, and metatheory for Quantifier. It is available at `github.com/antonyeagle/edl`.

tense example from §23.

Key Ideas in §38

- The two turnstiles are distinct concepts. But they coincide: every provable argument corresponds to an entailment, and *vice versa*.
- This means that we can pick and choose our methods to suit our task. If we want to show an entailment, we can sometimes most effectively proceed by providing a proof. And if we want to show something isn't provable, it can be more efficient to provide a countermodel.
- With the basics under your belt, there are many possible next steps in logic. It is natural to try to establish more metatheoretical results about the systems we have discussed, and some resources are indicated in the text to help you do this if you wish.

	Yes	**No**
Is $\mathcal{A}$ a **logical truth**?	give a proof which shows $\vdash \mathcal{A}$	give an interpretation in which $\mathcal{A}$ is false
Is $\mathcal{A}$ a **logical falsehood**?	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**?	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**?	give an interpretation in which all of $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ are true	prove a logical falsehood 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**?	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

39

Next Steps in Logic

Appendices

Appendix A

Alternative Terminology and Notation

Alternative terminology

Sentential logic The study of Sentential goes by other names. The name we have given it – sentential logic – derives from the fact that it deals with whole sentences as its most basic building blocks. Other features motivate different names. Sometimes it is called *truth-functional logic*, because it deals only with assignments of truth and falsity to sentences, and its connectives are all truth-functional. Sometimes it is called *propositional logic*, which strikes me as a misleading choice. This may sometimes be innocent, as some people use 'proposition' to mean 'sentence'. However, noting that different sentences can mean the same thing, many people use the term 'proposition' as a useful way of referring to the meaning of a sentence, what that sentence expresses. In *this* sense of 'proposition', 'propositional logic' is not a good name for the study of Sentential, since synonymous but not logically equivalent sentences like 'Vixens are bold' and 'Female foxes are bold' will be logically distinguished even though they express the same proposition.

Quantifier logic The study of Quantifier goes by other names. Sometimes it is called *predicate logic*, because it allows us to apply predicates to objects. Sometimes it is called *first-order logic*, because it makes use only of quantifiers over objects, and variables that can be substituted for constants. This is to be distinguished from *higher-order logic*, which introduces quantification over properties, and variables that can be substituted for predicates. (This device would allow us to formalise such sentences as 'Jane and Kane are alike in *every respect*', treating the italicised phrase as a quantifier over 'respects', i.e., properties. This results in something like $\forall P(Pj \leftrightarrow Pk)$, which is *not* a sentence of Quantifier.)

Atomic sentences Some texts call atomic sentences *sentence letters*. Many texts use lower-case roman letters, and subscripts, to symbolise atomic sentences.

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 make any important use of the contrasting class of 'ill-formed formulas'). I have stuck with 'formula'.

In §6, I defined *sentences* of Sentential. These are also sometimes called 'formulas' (or 'well-formed formulas') since in Sentential, unlike Quantifier, there is no distinction between a formula and a sentence.

Valuations Some texts call valuations *truth-assignments*; others call them *structures*.

Expressive adequacy Some texts describe Sentential as *truth-functionally complete*, rather than expressively adequate.

n-place predicates I 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', 'ternary', etc.

Names In Quantifier, I have used '*a*', '*b*', '*c*', for names. Some texts call these 'constants', because they have a constant referent in a given interpretation, as opposed to variables which have variable referents. 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'.

Interpretations Some texts call interpretations *models*; others call them *structures*.

Alternative notation

In the history of formal logic, different symbols have been used at different times and by different authors. Often, authors were forced to use notation that their printers could typeset.

This appendix presents some common symbols, so that you can recognise them if you encounter them in an article or in another book. Unless you are reading a research article in philosophical or mathematical logic, these symbols are merely different notations for the very same underlying things. So the truth-functional connective we refer to with '∧' is the very same one that another textbook might refer to with '&'. Compare: the number six can be referred to by the numeral '6', the Roman numeral 'VI', the English word 'six', the German word 'sechs', the kanji character '六', etc.

Negation Two commonly used symbols are the *not sign*, '$\neg$', and the *tilde operator*, '$\sim$'. 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 an overline to indicate negation, so that '$\overline{\mathcal{A}}$' expresses the same thing as '$\neg\mathcal{A}$'. This is clear enough if $\mathcal{A}$ is an atomic sentence, but quickly becomes cumbersome if we attempt to nest negations: '$\neg(A \wedge \neg(\neg\neg B \wedge C))$' becomes the unwieldy

$$\overline{A \wedge \overline{(\overline{\overline{B}} \wedge C)}}.$$

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 '*&*'.) 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, '$\cdot$', is used (you may have seen this in *Argument and Critical Thinking*). In some older texts, there is no symbol for conjunction at all; 'A and B' is simply written 'AB'. (These are often texts that use the overlining notation for negation. Such texts often involve languages in which conjunction and negation are the only connectives, and they typically also dispense with parentheses which are unnecessary in such austere languages.)

Material conditional There are two common symbols for the material conditional: the *arrow*, '$\rightarrow$', and the *hook*, '$\supset$'. Rarely you might see '$\Rightarrow$'.

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

Quantifiers The universal quantifier is typically symbolised '$\forall$' (a rotated 'A'), and the existential quantifier as '$\exists$' (a rotated 'E'). In some texts, there is no separate symbol for the universal quantifier. Instead, the variable is just written in parentheses in front of the formula that it binds. For example, they might write '$(x)Px$' where we would write '$\forall xPx$'.

The common alternative notations are summarised below:

negation	$\neg$, $\sim$, $\overline{\mathcal{A}}$
conjunction	$\wedge$, &, $\cdot$
disjunction	$\vee$
conditional	$\rightarrow$, $\supset$, $\Rightarrow$
biconditional	$\leftrightarrow$, $\equiv$
universal quantifier	$\forall x$, (x)

Doing without parentheses: Polish notation

We have established some conventions governing when you can omit parentheses in Sentential (in §§6.5 and 10.3). These conventions are useful in practice. It can be hard to read sentences with many nested parentheses. Moreover, it is easy to make a mistake when constructing sentences, if you accidentally omit a needed bracket. So there has been some interest in ways of doing sentential logic without parentheses. Frege's initial formulation of sentential logic in his *Begriffschrift* (1879) lacked parentheses, but involves a complicated nonlinear branching structure which is difficult to typeset and has not been much used by anyone since.

The now-standard approach to sentential logic without parentheses was due to the Polish logician Jan Łukasiewicz, and for that reason it has become known as POLISH NOTATION. It is a purely syntactic variant of our system Sentential: the truth-functions expressed by the connectives remain the same, but the sentences are written very differently.[1]

The basic idea is that rather than writing a connective in a logically complex sentence between the two subsentences, it should be written before them – in a sense, it treats all binary connectives like negation. So if $\mathcal{A}$ and $\mathcal{B}$ are two sentences, the sentence we write in Sentential as $\mathcal{A} \to \mathcal{B}$ would be written as $C\mathcal{A}\mathcal{B}$, where the 'C' is the symbol used for the conditional connective. Because the language uses some of the standard upper case letters for its connectives, there is potential for confusion if we use them for atomic sentences too. So the atomic sentences of this Polish language will be 'p', 'q', or 'r', with any numerical subscripts.

The standard connectives are these, with their Sentential equivalents:

Connective	Polish	Sentential
negation	$N\mathcal{A}$	$\neg\mathcal{A}$
conjunction	$K\mathcal{A}\mathcal{B}$	$(\mathcal{A} \wedge \mathcal{B})$
disjunction	$A\mathcal{A}\mathcal{B}$	$(\mathcal{A} \vee \mathcal{B})$
conditional	$C\mathcal{A}\mathcal{B}$	$(\mathcal{A} \to \mathcal{B})$
biconditional	$E\mathcal{A}\mathcal{B}$	$(\mathcal{A} \leftrightarrow \mathcal{B})$

We can recursively define a sentence of this Polish language:

1. Any atomic sentence (i.e.,$p, q, r, p_1, q_1, r_1, \ldots$) is a sentence;
2. If $\mathcal{A}$ and $\mathcal{B}$ are sentences, then so are '$N\mathcal{A}$', $K\mathcal{A}\mathcal{B}$, $A\mathcal{A}\mathcal{B}$, $C\mathcal{A}\mathcal{B}$ and $E\mathcal{A}\mathcal{B}$;
3. Nothing else is a sentence.

[1] In what follows I draw on Peter Simons (2017) 'Łukasiewicz's Parenthesis-Free or Polish Notation', in Edward N Zalta, ed., *The Stanford Encyclopedia of Philosophy*, https://plato.stanford.edu/archives/spr2017/entries/lukasiewicz/polish-notation.html.

It is easy to see that one can type sentences in Polish notation without the use of any special symbols on a standard typewriter keyboard.

The notation doesn't require parentheses. The ambiguous string '$P \to Q \to R$' may correspond to either of these two Sentential sentences: (i) '$(P \to (Q \to R))$' or (ii) $((P \to Q) \to R)$. These are symbolised in Polish notation as, respectively, (i′) '$CpCqr$' and (ii′) '$CCpqr$'. Why?

- You know, from the recursive definition, that any logically complex sentence is formed by taking a connective and placing one or two sentences after it. So the main connective is always the leftmost character.
- You then need only to identify a sentence that follows it. So in (i′), the main connective is 'C', which is followed by the atomic sentence 'p' and the complex sentence 'Cqr'; in (ii′), with the same main connective, the sentences which follow are 'Cpq' and 'r'.
- No ambiguity is possible. It would be possible only if there were some sentences $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ such that the string $\mathcal{AB}$ is identical to the string $\mathcal{CD}$, where $\mathcal{A} \neq \mathcal{C}$. That would mean, without loss of generality, that $\mathcal{A}$ is an initial part of $\mathcal{C}$. But then $\mathcal{C}$ cannot be a sentence, since it is a string comprised of a sentence plus some other symbols tacked on the end, which is not a sentence, by the recursive definition.

The notation can be very concise. Compare the following:

$$\Big(\big((P \land Q) \to R\big) \to \big(P \land (\neg Q \lor R)\big)\Big);$$

$$CCKpqrKpANqr.$$

The Sentential version has 22 characters, 10 of them parentheses; the Polish version just 12.

The notation never really caught on, partly because - as in the example above – it is not always immediate to the naked eye where one constituent sentence begins and another ends. But the main obstacle to its wider use was the lack of any easy way to indicate the scope of a quantifier. Thus the notation has become something of a historical curiosity.

Appendix B

Quick Reference

Schematic Truth Tables

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

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{A} \wedge \mathcal{B}$	$\mathcal{A} \vee \mathcal{B}$	$\mathcal{A} \rightarrow \mathcal{B}$	$\mathcal{A} \leftrightarrow \mathcal{B}$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

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	$(P \rightarrow Q)$
P only if Q	$(P \rightarrow Q)$
P if and only if Q	$(P \leftrightarrow Q)$
P unless Q	$(P \vee Q)$

Predicates

All Fs are Gs	$\forall x(Fx \rightarrow Gx)$
Some Fs are Gs	$\exists x(Fx \wedge Gx)$
Not all Fs are Gs	$\neg\forall x(Fx \rightarrow Gx)$ or $\exists x(Fx \wedge \neg Gx)$
No Fs are Gs	$\forall x(Fx \rightarrow \neg Gx)$ or $\neg\exists x(Fx \wedge Gx)$

Identity

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

Using identity to symbolize quantities

There are at least _____ Fs.

one: $\exists x Fx$
two: $\exists x_1 \exists x_2 (Fx_1 \wedge Fx_2 \wedge \neg x_1 = x_2)$
three: $\exists x_1 \exists x_2 \exists x_3 (Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge \neg x_1 = x_2 \wedge \neg x_1 = x_3 \wedge \neg x_2 = x_3)$
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: $\forall x_1 \forall x_2 \big((Fx_1 \wedge Fx_2) \rightarrow x_1 = x_2\big)$
two: $\forall x_1 \forall x_2 \forall x_3 \big((Fx_1 \wedge Fx_2 \wedge Fx_3) \rightarrow (x_1 = x_2 \vee x_1 = x_3 \vee x_2 = x_3)\big)$
three: $\forall x_1 \forall x_2 \forall x_3 \forall x_4 \big((Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge Fx_4) \rightarrow$
$(x_1 = x_2 \vee x_1 = x_3 \vee x_1 = x_4 \vee x_2 = x_3 \vee x_2 = x_4 \vee x_3 = x_4)\big)$
n: $\forall x_1 \ldots \forall x_{n+1} \big((Fx_1 \wedge \ldots \wedge Fx_{n+1}) \rightarrow (x_1 = x_2 \vee \ldots \vee x_n = x_{n+1})\big)$

There are exactly _____ Fs.

One way to say 'there are exactly n Fs' is to conjoin two of the symbolizations above and say 'there are at least n Fs and there are at most n Fs.' The following equivalent formulae are shorter:

zero: $\forall x \neg Fx$
one: $\exists x \big(Fx \wedge \forall y(Fy \rightarrow x = y)\big)$
two: $\exists x_1 \exists x_2 \big(Fx_1 \wedge Fx_2 \wedge \neg x_1 = x_2 \wedge \forall y\big(Fy \rightarrow (y = x_1 \vee y = x_2)\big)\big)$
three: $\exists x_1 \exists x_2 \exists x_3 \big(Fx_1 \wedge Fx_2 \wedge Fx_3 \wedge \neg x_1 = x_2 \wedge \neg x_1 = x_3 \wedge \neg x_2 = x_3 \wedge$
$\forall y\big(Fy \rightarrow (y = x_1 \vee y = x_2 \vee y = x_3)\big)\big)$
n: $\exists x_1 \ldots \exists x_n \big(Fx_1 \wedge \ldots \wedge Fx_n \wedge \neg x_1 = x_2 \wedge \ldots \wedge \neg x_{n-1} = x_n \wedge$
$\forall y\big(Fy \rightarrow (y = x_1 \vee \ldots \vee y = x_n)\big)\big)$

Basic deduction rules for Sentential

Conjunction Introduction, p. 244

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

Conjunction Elimination, p. 246

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

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

Conditional Introduction, p. 257

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

Conditional Elimination, p. 249

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

Negation Introduction, p. 270

i	| $\mathcal{A}$	
j	| $\mathcal{B}$	
k	| $\neg\mathcal{B}$	
	$\neg\mathcal{A}$	$\neg$I i–j, i–k

Negation Elimination, p. 272

i	| $\neg\mathcal{A}$	
j	| $\mathcal{B}$	
k	| $\neg\mathcal{B}$	
	$\mathcal{A}$	$\neg$E i–j, i–k

Disjunction Introduction, p. 252

m	$\mathcal{A}$	
	$\mathcal{A} \lor \mathcal{B}$	$\lor$I m

m	$\mathcal{A}$	
	$\mathcal{B} \lor \mathcal{A}$	$\lor$I m

Disjunction Elimination, p. 267

m	$\mathcal{A} \lor \mathcal{B}$	
i	| $\mathcal{A}$	
j	| $\mathcal{C}$	
k	| $\mathcal{B}$	
l	| $\mathcal{C}$	
	$\mathcal{C}$	$\lor$E m, i–j, k–l

Biconditional Introduction, p. 265

i	| $\mathcal{A}$	
j	| $\mathcal{B}$	
k	| $\mathcal{B}$	
l	| $\mathcal{A}$	
	$\mathcal{A} \leftrightarrow \mathcal{B}$	$\leftrightarrow$I i–j, k–l

Biconditional Elimination, p. 250

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

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

Reiteration, p. 253

m	$\mathcal{A}$	
	$\vdots$	
	$\mathcal{A}$	R m

Derived rules for Sentential, §33

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

Modus Tollens

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

Double Negation Elimination

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

Tertium non datur

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

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

Basic deduction rules for Quantifier

Universal elimination, p. 303

m	$\forall x\mathcal{A}$	
	$\mathcal{A}\vert_{c\curvearrowright x}$	$\forall$E m

c can be any name

Universal introduction, p. 308

m	$\mathcal{A}\vert_{c\curvearrowright x}$	
	$\forall x\mathcal{A}$	$\forall$I m

c must not occur in any undischarged assumption, or in $\mathcal{A}$

Identity introduction, p. 328

	$c = c$	=I

Existential introduction, p. 305

m	$\mathcal{A}\vert_{c\curvearrowright x}$	
	$\exists x\mathcal{A}$	$\exists$I m

c can be any name

Existential elimination, p. 312

m	$\exists x\mathcal{A}$	
i	$\quad\mathcal{A}\vert_{c\curvearrowright x}$	
j	$\quad\mathcal{B}$	
	$\mathcal{B}$	$\exists$E m, i–j

c must not occur in any undischarged assumption, in $\exists x\mathcal{A}$, or in $\mathcal{B}$

Identity elimination, p. 329

m	$a = b$	
n	$\mathcal{A}\vert_{a\curvearrowright x}$	
	$\mathcal{A}\vert_{b\curvearrowright x}$	=E m, n

Derived rules for Quantifier, §36

m	$\forall x\neg\mathcal{A}$	
	$\neg\exists x\mathcal{A}$	$\mathrm{CQ}_{\forall/\neg\exists}$ m

m	$\neg\exists x\mathcal{A}$	
	$\forall x\neg\mathcal{A}$	$\mathrm{CQ}_{\neg\exists/\forall}$ m

m	$\exists x\neg\mathcal{A}$	
	$\neg\forall x\mathcal{A}$	$\mathrm{CQ}_{\exists/\neg\forall}$ m

m	$\neg\forall x\mathcal{A}$	
	$\exists x\neg\mathcal{A}$	$\mathrm{CQ}_{\neg\forall/\exists}$ m

Alternative identity elimination, p. 330

m	$a = b$	
n	$\mathcal{A}\vert_{b\curvearrowright x}$	
	$\mathcal{A}\vert_{a\curvearrowright x}$	=ES m, n

Appendix C

Index of defined terms

active assumption, 240
antecedent, 44
antisymmetric, 195
argument, 2
assumption lines, 240
assumptions, 240
asymmetric, 195
atomic formulae, 172
atomic sentences, 33

biconditional, 45
biconditional elimination, 250
biconditional introduction, 265
binary relation, 193
bivalence, 90
bound by, 175
bound variable, 174

canonical clause, 27
clash of variables, 144
closed, 256
commutative, 67
complete truth table, 78
completeness, 336
compositional, 66
compound, 28, 66
conclusion, 2
conclusive, 6
conditional elimination, 249
conditional introduction, 258
conditional proof, 280
conjunction, 40
conjunction elimination, 246
conjunction introduction, 245
conjuncts, 40
consequent, 44
context sensitive, 76
contingent, 21
contradiction, 81
conversational context, 113
coordinator, 28
counterfactuals, 72
countermodel, 227

declarative sentences, 17
definite descriptions, 160
derived, 289
determiner phrase, 26
directed graph, 189
directed hypergraphs, 192
discharging, 256
disjunction, 43
disjunction elimination, 268
disjunction introduction, 252
disjunctive normal form, 108
disjunctive syllogism, 290
disjuncts, 43
domain, 119
double negation elimination, 292
double turnstile, 86
dummy pronoun, 117
dummy pronouns, 116

elimination, 239
empty, 125
empty extension, 181
entail, 85, 218
enthymeme, 14
epistemology, 9
equivalence classes, 197
equivalence relation, 196, 331
Euler diagram, 187
excluded middle, 273
exclusive or, 43
existential introduction, 305
existential quantifier, 118
exportation, 260
expression, 50, 171
extension, 179
extensional language, 180

False, 18
form, 10
formal language, 31
formal logic, 11
formal proof, 239
formation tree, 53
free choice 'any', 129
free variable, 174
function, 64

generic, 134, 160
grammatical subject, 28

identity of indiscernibles, 184
iff, 45
immediate subsentences, 66
importation, 260
impossibility, 20
inclusive or, 43
indefinite descriptions, 160
indicative, 72
indirect, 102
indiscernibility of identicals, 184
Induction, 13
Inductive logic, 13
inferentialist, 282
instantiating name, 207
interpretation, 185
intersective, 131
intransitive, 194
introduction, 239
invalid, 11, 219
irreflexive, 193

jointly consistent, 18, 83, 217
jointly contrary, 285
jointly formally consistent, 19
jointly inconsistent, 18, 83, 217

Leibniz' Law, 155
logical consequence, 3
logical falsehood, 81, 217
logical truth, 80, 217
logically equivalent, 81, 217

main connective, 53, 173
many-valued logics, 106
mention, 57
metalanguage, 58
metalinguistic negation, 166
modus ponens, 250
modus tollens, 291
monotonic, 23

names, 114
narrow scope, 165
natural deduction, 239
natural kind terms, 114
necessary falsehood, 20
necessary truth, 20
negation elimination, 272
negation introduction, 270
negative polarity (NPI) 'any', 129
nodes, 189
non-identity predicate, 154
nonsymmetric, 195
noun phrase, 26
numerical identity, 153

object language, 58
one-place, 137
opaque, 185
open sentences, 118
ordered pairs, 182
ordering, 197

paraphrases, 37

partial order, 197
partial truth table, 98
partition, 197
phrase structure grammars, 26
Polish notation, 344
predicate, 28
premises, 2
presupposition failure, 166
privative, 131
Proper names, 113
property, 186
provably equivalent, 285

qualitatively identical, 153
quantifier phrase, 118
quasi-canonical clauses, 31
quiver, 191

range, 240
recursive, 51
reductio, 230
reflexive, 154, 193
reiteration, 253
relation, 187
representationalism, 282

satisfies, 213
schematic truth table, 67
scope, 53, 173
semantic presupposition, 166
semantics, 2, 31
sentence, 51
sentence connectives, 28
sentences, 175
serial, 200
singular term, 112
sound, 12
soundness, 335
strict order, 197
strict total order, 197
structural words, 28
structure, 7, 25
subjunctive, 72
subproofs, 255
subsective, 133
subsentences, 27
subset, 187, 198
substituting, 207
substitution instance, 207
substitutional quantification, 204
symbolisation key, 33
symmetric, 154, 195
syntactic tree, 26
syntax, 31

tautologies, 80
term, 172
tertium non datur, 293
the conditional, 44
theorem, 284
three-place, 138
total, 197
total function, 75
total order, 197
transitive, 154, 194, 331
True, 18
truth table, 76
truth value, 17
truth-function, 65
truth-functional, 66
truth-functionally complete, 106
two-place, 138

undischarged assumptions, 240
universal elimination, 304
universal quantifier, 118
use, 57

valid, 11, 219
valuation, 75
variable, 118
variable assignment, 211
verb phrase, 26
verb phrases, 115

wide scope, 165

Acknowledgements

Antony Eagle would like to thank P.D. Magnus and Tim Button for their work from which this text derives, and those acknowledged below who helped them. He also wishes to thank Atheer Al-Khalfa, Caitlin Bettess, Andrew Carter, Keith Dear, Jack Garland, Bowen Jiang, Millie Lewis, Yaoying Li, Jon Opie, Matt Nestor, Jaime von Schwarzburg, and Mike Walmer for comments on successive versions of the Adelaide text.

Tim Button would like to thank P.D. Magnus for his extraordinary act of generosity, in making `forall`x 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.

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.

About the Authors

Antony Eagle is Associate Professor of Philosophy at the University of Adelaide. His research interests include metaphysics, philosophy of probability, philosophy of physics, and philosophy of logic and language. `antonyeagle.org`

Tim Button is a Lecturer in Philosophy at UCL. His first book, *The Limits of Realism*, was published by Oxford University Press in 2013. `www.homepages.ucl.ac.uk/~uctytbu/index.html`

P.D. Magnus is a professor at the University at Albany, State University of New York. His primary research is in the philosophy of science. `www.fecundity.com/job/`

In the Introduction to his *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.