forall$x$

BRISTOL

Version from September 24, 2021
Download here.
Please redownload every week.

forall x: Bristol

Textbook for
PHIL10032: Logic and Critical Thinking 2020

Also used in
PHIL10014: Introduction to Formal Logic 2020

By **P. D. Magnus**
Tim Button
with additions by
J. Robert Loftis
Robert Trueman
remixed and revised by
Aaron Thomas-Bolduc
Richard Zach
and by
Catrin Campbell-Moore

Last updated September 24, 2021

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

Aaron Thomas-Bolduc & Richard Zach
University of Calgary

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

Tim Button
University of Cambridge

used under a CC BY-SA 3.0 license, which is based in turn on `forall`x, by

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

used under a CC BY-SA 3.0 license.
and was remixed, revised, & expanded by

Aaron Thomas-Bolduc & Richard Zach
University of Calgary
then altered by

Catrin Campbell-Moore
University of Bristol Further contributors: Stuart Presnell, Richard Pettigrew.

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

The LaTeX source for this book is available on GitHub.

Preface

The book is divided into nine parts. Part I introduces the topic and notions of logic in an informal way, without introducing a formal language yet. Parts II–IV concern truth-functional languages. In it, sentences are formed from basic sentences using a number of connectives ('or', 'and', 'not', 'if ... then') which just combine sentences into more complicated ones. We discuss logical notions such as entailment in two ways: semantically, using the method of truth tables (in Part III) and proof-theoretically, using a system of formal derivations (in Part IV). Parts V–VII deal with a more complicated language, that of first-order logic. It includes, in addition to the connectives of truth-functional logic, also names, predicates, identity, and the so-called quantifiers. These additional elements of the language make it much more expressive than the truth-functional language, and we'll spend a fair amount of time investigating just how much one can express in it. Again, logical notions for the language of first-order logic are defined semantically, using interpretations, and proof-theoretically, using a more complex version of the formal derivation system introduced in Part IV. Part IX covers two advanced topics: that of conjunctive and disjunctive normal forms and the expressive completeness of the truth-functional connectives, and the soundness of natural deduction for TFL. Part X discusses the extension of TFL by non-truth-functional operators for possibility and necessity: modal logic.

In the appendices you'll find a discussion of alternative notations for the languages we discuss in this text, of alternative derivation systems, and a quick reference listing most of the important rules and definitions. The central terms are listed in a glossary at the very end.

This book is based on a text originally written by P. D. Magnus in the version revised and expanded by Tim Button. It also includes some material (mainly exercises) by J. Robert Loftis. The material in Part X is based on notes by Robert Trueman, and the material in Part ?? on two chapters from Tim Button's open text *Metatheory*. Aaron Thomas-Bolduc and Richard Zach have combined elements of these texts into the present version, changed some of the terminology and examples, rewritten some sections, and added material of their own. Catrin Campbell-Moore has then made very minor alterations for the Bristol course. The resulting text is licensed under a Creative Commons Attribution-ShareAlike 4.0 license.

Contents

PART I

Arguments

CHAPTER 1

Arguments

1.1 Introduction

Much of philosophical practice is about argument and analysis. Arguing in support of or against some position, or understanding someone else's argument. Logic is the study of the practice of argument and analysis, abstracted from the specific details of a particular case.

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

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

- If I acted of my own free will, then I could have acted otherwise.
- I could not have acted otherwise.
- Therefore: I did not act of my own free will.

We here have a series of sentences which may either be true or false. The final sentence, "I did not act of my own free will." expresses the *conclusion* of the argument. The two sentences before that are the *premises* of the argument. In a good argument, the conclusion follows from the premises. If you believe the premises then the argument should lead you to believing the conclusion.

Logic provides the ideal model of good argument: rational argument without rhetoric. The logical study of an argument can show whether it supports its conclusion or is flawed. Logic focuses only on the statements presented and the relationships between them. Extraneous factors are set aside: unspoken assumptions, additional connotations of words, appeal to emotions.

Logical thinking can help us to work out the intended interpretation of a text, and to find alternative unintended interpretations. This can be helpful when reading someone else's writing, and essential when we are trying to write unambiguously. Logical analysis can help us to find ambiguity and alternative interpretations, and to write in a precise and unambiguous way that can only be interpreted as we intend. These are vital skills used in all philosophy as well as in life more generally.

This Part discusses arguments in natural languages like English. Throughout this textbook we will also consider arguments in formal languages and say what it is for those to be valid or invalid. We want formal validity, as defined in the formal language, to have at least some of the important features of natural-language validity.

1.2 Finding the components of an argument

Arguments consist of a list of *premises* along with a *conclusion*. In a good argument, the conclusion will follow from the premises. Our standard way to present them is:

- Premise 1
- Premise 2
- ...
- Premise n
- Therefore: Conclusion

For example

- If I acted of my own free will, then I could have acted otherwise.
- I could not have acted otherwise.
- Therefore: Therefore: I did not act of my own free will.

Often arguments are presented simply in a paragraph of text, or in a speech or article, and we first have to work out what the premises and conclusions are. Sometimes it's easy, for example:

> If I acted of my own free will, then I could have acted otherwise. But, I could not have acted otherwise. So, I did not act of my own free will.

But often it is a significant piece of work to work out the premises and conclusion of an argument.

Many arguments start with premises, and end with a conclusion, but not all of them. It might start with the conclusion:

> **We should not have a second Brexit referendum**. A second Brexit referendum would erode the very basis of democracy by suggesting that rule by the majority is an insufficient condition for democratic legitimacy.

Or it might have been presented with the conclusion in the middle:

> Since the first Brexit referendum was made under false pretences, **the voters deserve a further say on any final deal agreed with Brussels**. After all, decisions as big as this need to have the public support, which has to come from a referendum.

Sometimes premises or the conclusion may be clauses in a sentence. A complete argument may even be contained in a single sentence:

> The butler has an alibi; so they cannot have done it.

This argument has one premise, followed immediately by its conclusion.

One particular kind of sentence can be confusing. Consider:

- If the murder weapon was a gun, then Prof. Plum did it.

These conditional, or “if-then”, statements might look like it expresses the argument, but in itself it does not. It’s just stating a fact, albeit a conditional fact. It might also be used in an argument, even as the conclusion of the argument:

- If I have free will, then there is some event that I could have caused to go differently.
- If determinism is true, then there is no event that I could have caused to go differently.
- Therefore: If determinism is true, I do not have free will.

As a guideline, there are some words you can look for which are often used to indicate whether something is a premise or conclusion:

Words often used to indicate an argument's conclusion:

so, therefore, hence, thus, accordingly, consequently

Words often used to indicate a premise:

since, because, as, given that, recalling that, after all

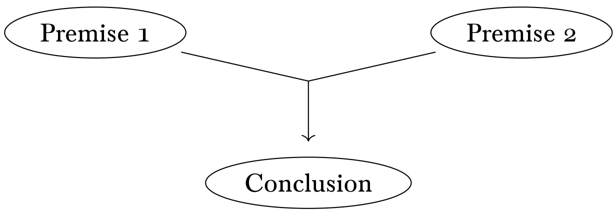
In analysing an argument, there is no substitute for a good nose. Whenever you come across an argument in a piece of philosophy you read, be it lecture notes, primary text, or secondary text, or in a newspaper article or on the internet, practice identifying the premises and conclusion.

Sometimes, though, people aren't giving arguments but are simply presenting facts or stating their opinion. For example, the following do not contain arguments, they're not trying to convince us of anything.

- I don't like cats. I think they're evil.
- Hundreds of vulnerable children as young as 10, who have spent most of their lives in the UK, are having their applications for British citizenship denied for failing to pass the government's controversial 'good character' test.

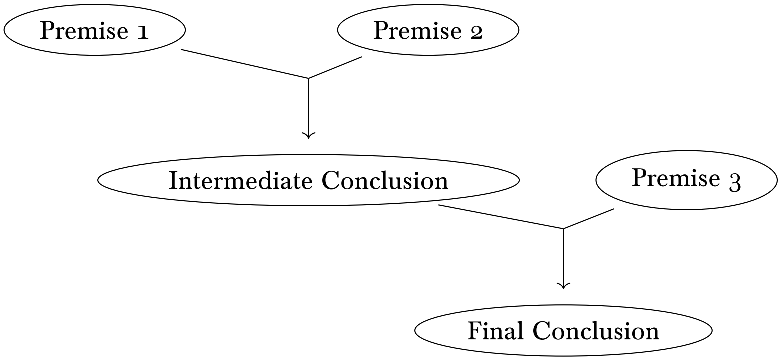
1.3 Intermediate Conclusions

We said an argument is given by a collection of *premises* along with a single *conclusion*. We might represent this as something like:



The premises are working together to lead to the conclusion.

But sometimes in the process of someone making an argument someone will make use of *intermediate conclusions*. Such arguments might have a structure more like:



However, we say that an argument is only something of the first kind. So what do we say about the second kind of thing? We can consider it two ways. We might could consider it as an argument from premise 1, 2 and 3 to the conclusion. Or alternatively we can think of it as two arguments of the first kind chained together, one from premise 1 and 2 to the intermediate conclusion, and the second from the intermediate conclusion and premise 3 to the final conclusion.

1.4 Sentences

What kinds of things are the premises and conclusions of arguments? They are sentences which can either be true or false. Such sentences are called DECLARATIVE SENTENCES.

There are many other kinds of sentences, for example:

Questions 'Are you sleepy yet?' is an interrogative sentence. Although you might be sleepy or you might be alert, the question itself is neither true nor false. For this reason, questions will not count as declarative sentences. Suppose you answer the question: 'I am not sleepy.' This is either true or false, and so it is a declarative sentences. Generally, *questions* will not count as declarative sentences, but *answers* will.

'What is this course about?' is not a declarative sentence (in our sense). 'No one knows what this course is about' is a declarative sentence.

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

Exclamations 'Ouch!' is sometimes called an exclamatory sentence, but it is not the sort of thing which is true or false. 'That hurt!', however, is a declarative sentence.

Our focus is only on *declarative sentences* — those sentences which can be true or false — for example 'spiders have eight legs'. We typically drop the term 'declarative' and simply call them sentences, but bear in mind that it is only these sorts of sentences that are relevant in this textbook.

You should not confuse the idea of a sentence that can be true or false with the difference between fact and opinion. Often, sentences in logic will express things that would count as facts— such as 'spiders have eight legs' or 'Kierkegaard liked almonds.' They can also express things that you might think of as matters of opinion—such as, 'Almonds are tasty.' In other words, a sentence is not disqualified from being part of an argument because we don't know if it is true or false, or because its truth or falsity is a matter of opinion. All that matters is whether it is the sort of thing that could be true or false. If it is, it can play the role of premise or conclusion.

When you are reading a text and putting it in our standard form you should make sure that your premises and conclusions are declarative sentences. You should also make them as clear as possible. Each premise and the conclusion should be able to be read and understood independently. Any context from the original paragraph should be copied over to each of the premises and conclusions. For example:

> Donating to charity no strings attached is the most effective way to do so. So if you are going to donate to charity, you should do it this way.

When presenting this we should fill out "this way" with the relevant way. So I'd write:

- Donating to charity no strings attached is the most effective way to do so.
- Therefore: If you are going to donate to charity, you should do so no strings attached.

1.5 Consequence and validity

In §1, we talked about arguments, i.e., a collection of sentences (the premises), followed by a single sentence (the conclusion). We said that some words, such as "therefore," indicate which sentence in is supposed to be the conclusion. "Therefore," of course, suggests that there is a connection between the premises and the conclusion, namely that the conclusion *follows from*, or *is a consequence of*, the premises.

This notion of consequence is one of the primary things logic is concerned with. One might even say that logic is the science of what follows from what. Logic develops theories and tools that tell us when a sentence follows from some others.

What about the following argument:

- Either the butler or the gardener did it.
- The butler didn't do it.
- Therefore: The gardener did it.

We don't have any context for what the sentences in this argument refer to. Perhaps you suspect that "did it" here means "was the perpetrator" of some unspecified crime. You might imagine that the argument occurs in a mystery novel or TV show, perhaps spoken by a detective working through the evidence. But even without having any of this information, you probably agree that the argument is a good one in the sense that whatever the premises refer to, if they are both true, the conclusion cannot but be true as well. If the first premise is true, i.e., it's true that "the butler did it or the gardener did it," then at least one of them "did it," whatever that means. And if the second premise is true, then the butler did not "do it." That leaves only one option: "the gardener did it" must be true. Here, the conclusion follows from the premises. We call arguments that have this property VALID.

By way of contrast, consider the following argument:

- If the driver did it, the maid didn't do it.
- The maid didn't do it.
- Therefore: The driver did it.

We still have no idea what is being talked about here. But, again, you probably agree that this argument is different from the previous one in an important respect. If the premises are true, it is not guaranteed that the conclusion is also true. The premises of this argument do not rule out, by themselves, that someone other than the maid or the driver "did it." So there is a case where both premises are true, and yet the driver didn't do it, i.e., the conclusion is not true. In this second argument, the conclusion does not follow from the premises. If, like in this argument, the conclusion does not follow from the premises, we say it is INVALID.

1.6 Validity

How did we determine that the second argument is invalid? We pointed to a case in which the premises are true and in which the conclusion is not. This was the scenario where neither the driver nor the maid did it, but some third person did. Let's call such a case a COUNTEREXAMPLE to the argument. If there is a counterexample to an argument, the conclusion cannot be a consequence of the premises. For the conclusion to be a consequence of the premises, the truth of the premises must guarantee the truth of the conclusion. If there is a counterexample, the truth of the premises does not guarantee the truth of the conclusion.

As logicians, we want to be able to determine when the conclusion of an argument follows from the premises. And the conclusion is a consequence of the premises if there is no counterexample—no case where the premises are all true but the conclusion is not. This motivates a definition:

> A sentence Y is a CONSEQUENCE of sentences $X_1, \ldots, X_n$ if and only if there is no case where $X_1, \ldots, X_n$ are all true and Y is not true. (We then also say that Y FOLLOWS FROM $X_1, \ldots, X_n$ or that $X_1, \ldots, X_n$ ENTAIL Y.)

We said that arguments where the conclusion is a consequence of the premises are called valid, and those where the conclusion isn't a consequence of the premises are invalid. That is:

> An argument is VALID if and only if there is no case where all the premises are true and the conclusion false.

> An argument is INVALID if and only if it is not valid. That is, there is some case where all the premises are true and the conclusion is false.

1.7 Cases and types of validity

The “definitions” from the previous section are incomplete: it does not tell us what a “case” is or what it means to be “true in a case.” So far we’ve only seen an example: a hypothetical scenario involving three people. Of the three people in the scenario—a driver, a maid, and some third person—the driver and maid didn’t do it, but the third person did. In this scenario, as described, the driver didn’t do it, and so it is a case in which the sentence “the driver did it” is not true. The premises of our second argument are true, but the conclusion is not true: the scenario is a counterexample.

One thing that logicians do is to make the notion of “case” more precise, and investigate which arguments are valid when “case” is made precise in one way or another. If we take “case” to mean “hypothetical scenario” like the counterexample to the second argument, it’s clear that the first argument counts as valid. If we imagine a scenario in which either the butler or the gardener did it, and also the butler didn’t do it, we are automatically imagining a scenario in which the gardener did it. So any hypothetical scenario in which the premises of our first argument are true automatically makes the conclusion of our first argument true. This makes the first argument valid.

Making “case” more specific by interpreting it as “hypothetical scenario” is an advance. But it is not the end of the story. The first problem is that we don’t know what to count as a hypothetical scenario. Are they limited by the laws of physics? By what is conceivable, in a very general sense? What answers we give to these questions determine which arguments we count as valid.

Suppose the answer to the first question is “yes.” Consider the following argument:

- The spaceship *Rocinante* took six hours to reach Jupiter from Tycho space station.
- Therefore: The distance between Tycho space station and Jupiter is less than 14 billion kilometers.

A counterexample to this argument would be a scenario in which the *Rocinante* makes a trip of over 14 billion kilometers in 6 hours, exceeding the speed of light. Since such a scenario is incompatible with the laws of physics, there is no such scenario if hypothetical scenarios have to conform to the laws of physics. If hypothetical scenarios are not limited by the laws of physics, however, there is a counterexample: a scenario where the *Rocinante* travels faster than the speed of light.

Suppose the answer to the second question is "yes," and consider another argument:

- Priya is an ophthalmologist.
- Therefore: Priya is an eye doctor.

If we're allowing only conceivable scenarios, this is also a valid argument. If you imagine Priya being an ophthalmologist, you thereby imagine Priya being an eye doctor. That's just what "ophthalmologist" and "eye doctor" mean. A scenario where Priya is an ophthalmologist but not an eye doctor is ruled out by the conceptual connection between these words.

When we consider cases of various kinds in order to evaluate the validity of an argument, we will make a few assumptions. The first assumption is that every case makes every sentence true or false—at least, every sentence in the argument under consideration. So imagined scenarios have to specify all relevant facts. Any imagined scenario which leaves it undetermined if a sentence in our argument is true will not be considered as a potential counterexample.

Depending on what kinds of cases we consider as potential counterexamples, then, we arrive at different notions of consequence and validity. We might call an argument NOMOLOGICALLY VALID if there are no counterexamples that don't violate the laws of nature, and an argument CONCEPTUALLY VALID if there are no counterexamples that don't violate conceptual connections between words. For both of these notions of validity, aspects of the world (e.g., what the laws of nature are) and aspects of the meaning of the sentences in the argument (e.g., that "ophthalmologist" just means a kind of eye doctor) figure into whether an argument is valid.

One distinguishing feature of *logical* consequence, however, is that it should not depend on the content of the premises and conclusion, but only on their logical form. In other words, as logicians we want to develop a theory that can make finer-grained distinctions still. For instance, both

- Either Priya is an ophthalmologist or a dentist.
- Priya isn't a dentist.
- Therefore: Priya is an eye doctor.

and

- Either Priya is an ophthalmologist or a dentist.

- Priya isn't a dentist.
- Therefore: Priya is an ophthalmologist.

are valid arguments. But while the validity of the first depends on the content (i.e., the meaning of "ophthalmologist" and "eye doctor"), the second does not. The second argument is FORMALLY VALID. We can describe the "form" of this argument as a pattern, something like this:

- Either a is an F or a G.
- a isn't an F.
- Therefore: a is a G.

Here, a, F, and G are placeholders for appropriate expressions that, when substituted for a, F, and G, turn the pattern into an argument consisting of sentences. For instance,

- Either Mei is a mathematician or a botanist.
- Mei isn't a botanist.
- Therefore: Mei is a mathematician.

is an argument of the same form, but the first argument above is not: we would have to replace F by different expressions (once by "ophthalmologist" and once by "eye doctor") to obtain it from the pattern.

Moreover, the first argument is not formally valid. *Its* form is this:

- Either a is an F or a G.
- a isn't an F.
- Therefore: a is a H.

In this pattern we can replace F by "ophthalmologist" and H by "eye doctor" to obtain the original argument. But here is another argument of the same form:

- Either Mei is a mathematician or a botanist.
- Mei isn't a botanist.
- Therefore: Mei is an acrobat.

This argument is clearly not valid, since we can imagine a mathematician named Mei who is not an acrobat.

Our strategy as logicians will be to come up with a notion of "case" on which an argument turns out to be valid if it is formally valid. Clearly such a notion of "case" will have to violate not just some laws of nature but some laws of English. Since the first argument is invalid in this sense, we must allow as counterexample a case where Priya is an ophthalmologist but not an eye doctor. That case is not a conceivable situation: it is ruled out by the meanings of "ophthalmologist" and "eye doctor."

1.8 Sound arguments

Before we go on and execute this strategy, a few clarifications. Arguments in our sense, as conclusions which (supposedly) follow from premises, are of course used all the time in everyday, philosophical and scientific discourse. When they are, arguments are given to support or even prove their conclusions. Now, if an argument is valid, it will support its conclusion, but *only if* its premises are all true. Validity rules out the possibility that the premises are true and the conclusion is not true at the same time. It does not, by itself, rule out the possibility that the conclusion is not true, period. In other words, it is perfectly possibly for a valid argument to have a conclusion that isn't true!

Consider this example:

- Oranges are either fruit or musical instruments.
- Oranges are not fruit.
- Therefore: Oranges are musical instruments.

The conclusion of this argument is ridiculous. Nevertheless, it follows from the premises. *If* both premises are true, *then* the conclusion just has to be true. So the argument is valid.

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.
- Therefore: 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 no longer be true, even though both of the premises would remain true. Thus, there is a case where the premises of this argument are true without the conclusion being true. So the argument is invalid.

The important thing to remember is that validity is not about the actual truth or falsity of the sentences in the argument. It is about whether it is *possible* for all the premises to be true and the conclusion to be not true at the same time (in some hypothetical case). What is in fact the case has no special role to play; and what the facts are does not determine whether an argument is valid or not.[1] Nothing about the way things are can by itself determine if an argument is valid. It is often said that logic doesn't care about feelings. Actually, it doesn't care about facts, either.

When we use an argument to prove that its conclusion *is true*, then, we need two things. First, we need the argument to be valid, i.e., we need the conclusion to follow from the premises. But we also need the premises to be true. We will say that an argument is SOUND if and only if it is both valid and all of its premises are true.

The flip side of this is that when you want to rebut an argument, you have two options: you can show that (one or more of) the premises are not true, or you can show that the argument is not valid. Logic, however, will only help you with the latter!

1.9 Missing premises

If someone you disagree with makes an invalid argument, you might be tempted just to dismiss it as obviously incorrect. But it's more useful (and more charitable) to consider whether there are missing premises that could be filled in that make the argument better. Perhaps the author was assuming that this premise was so obvious that it didn't need to be stated.

For example an author might make the following argument:

- I could not have acted otherwise.
- Therefore: I did not act of my own free will.

This argument is invalid. But, it can be made valid by addition of the premise:

- If I could not have acted otherwise, I did not act of my own free will.

But be careful when you're filling in 'missing' premises. The aim is to help improve the argument, to make it more convincing, so you can assess it fairly. Only add extra premises that seem reasonable, or that you think the original author would agree with. There's no point in adding absurd or unreasonable premises, or premises that the

[1] Well, there is one case where it does: if the premises are in fact true and the conclusion is in fact not true, then we live in a counterexample; so the argument is invalid.

author wouldn't endorse. Then you just create a *strawman* argument – a caricature of the original argument.

> "Just how charitable are you supposed to be when criticizing the views of an opponent? If there are obvious contradictions in the opponent's case, then of course you should point them out, forcefully. If there are somewhat hidden contradictions, you should carefully expose them to view—and then dump on them. But the search for hidden contradictions often crosses the line into nitpicking, sea-lawyering, and—as we have seen—outright parody. The thrill of the chase and the conviction that your opponent has to be harboring a confusion somewhere encourages uncharitable interpretation, which gives you an easy target to attack. But such easy targets are typically irrelevant to the real issues at stake and simply waste everybody's time and patience, even if they give amusement to your supporters."
> *Daniel C. Dennett (2013). "Intuition Pumps And Other Tools for Thinking".*

Dennett formulates the following four rules (named after Anatol Rapoport) for "how to compose a successful critical commentary":

1. You should attempt to re-express your target's position so clearly, vividly, and fairly that your target says, "Thanks, I wish I'd thought of putting it that way."

2. You should list any points of agreement (especially if they are not matters of general or widespread agreement).

3. You should mention anything you have learned from your target.

4. Only then are you permitted to say so much as a word of rebuttal or criticism

1.10 Ampliative Arguments

Sometimes there are no plausible missing premises you could add to someone's argument to make it valid. However, this doesn't necessarily mean that the author was wrong or mistaken. Deductively valid arguments with plausible premises are good arguments, but they aren't the only good arguments there are. This is just as well, since many arguments we give in our everyday lives are not deductively valid, even after filling in plausible missing premises. Here's an example:

- 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.
- Therefore: It rains every January in London.

This argument generalises from observations about several cases to a conclusion about all cases—in each year listed, it rained in January, so it does in every year. Such arguments are called INDUCTIVE arguments. The argument could be made stronger by adding additional premises before drawing the conclusion: In January 2001, it rained in London; In January 2002.... But, however many premises of this form we add, the argument will remain invalid. Even if it has rained in London in every January thus far, it remains possible that London will stay dry next January. The point of all this is that inductive arguments—even good inductive arguments—are not (deductively) valid. They are not watertight. The premises might make the conclusion very likely, but they don't absolutely guarantee its truth. Unlikely though it might be, it is possible for their conclusion to be false, even when all of their premises are true.

Inductive arguments of the sort just given belong to a species of argument called AMPLIATIVE ARGUMENTS. This means that the conclusion goes beyond what you find in the premises. That is, the premises don't guarantee, or entail, the conclusion. They do, however, provide some support for it. These arguments are deductively invalid. They may be good and useful, however it is important to know the difference.

In this book, we will set aside the question of what makes for a good ampliative argument and focus instead on sorting the deductively valid arguments from the deductively invalid ones. But we pause here to mention some further forms of ampliative argument.

Inductive arguments, like the one we saw above, allow one to infer from a series of observed cases to a generalization that covers them: from all observed Fs have been Gs, we infer all Fs are Gs. We use these all the time. Every time I've drunk water from my tap, it's quenched my thirst; therefore, every time I ever drink water from my tap, it will quench my thirst. Every time I've stroked my neighbour's cat, it hasn't bitten me; therefore, every time I ever stroke my neighbour's cat, it won't bite me. And it's a form of arguments much beloved by scientists. Every time we've measured the acceleration of a body falling, it's matched Newton's theory, therefore, all bodies are governed by Newton's theory. The premises of these argument seem to make their conclusions likely without guaranteeing them. The areas of philosophy called inductive logic or confirmation theory try to make precise what that means and why it's true. And of course inductive arguments can go wrong. Before I visited Australia, every swan I'd every seen was white, and so I concluded that all swans were white; but when I visited Australia, I realised my conclusion was wrong, because some swans there are black.

A closely related, but different form of argument, is STATISTICAL. Here, we start with an observation about the proportion of Fs that are Gs in a sample that we've observed, and we infer that the same proportion of Fs are Gs in general. So, for instance, if I poll 1,000 people in Scotland eligible to vote in a second independence referendum, and 600 say that they'll vote yes, I might infer that 60% of all eligible voters will vote yes. Or if I test 1,000,000 people in England for an active infection, and 20,000 test positive, I might infer that 2% of the whole population has an active infection. How good these argument are depends on a number of things, and these are studied by statisticians. For instance, suppose you picked the 1,000 Scottish voters entirely at random from an anonymised version of the electoral register. But suppose that, when you deanonymised, you learned that, by chance, all of the people you'd picked were over 65, or they all lived on the Isle of Skye. Then you might worry that your sample, though random, was unrepresentative of the population as a whole. This question is a genuine concern for randomised controlled trials in medicine.

Abductive arguments provide an inference from a phenomenon you've observed to the *best explanation* of that phenomenon: from E, and the best explanation of E is H, you might conclude H. Again, this is extremely widespread. A classic sort of example would be the inferences that detectives draw during their investigations. They look at the evidence and the possible explanations of it, and they tend to conclude in favour of the best one. And similarly for doctors looking at a patient's suite of symptoms and trying to discover what ails them. Another important example comes from science. Here is Charles Darwin explaining what convinces him of his theory of natural selection:

> "It can hardly be supposed that a false theory would

explain, in so satisfactory a manner as does the theory of natural selection, the several large classes of facts above specified. It has recently been objected that this is an unsafe method of arguing; but it is a method used in judging of the common events of life, and has often been used by the greatest natural philosophers."
(Charles Darwin, On the origin of species by means of natural selection (6th ed.). London: John Murray)

CHAPTER 2

Other logical notions

In §??, we introduced the ideas of consequence and of valid argument. This is one of the most important ideas in logic. In this section, we will introduce are some similarly important ideas. They all rely, as did validity, on the idea that sentences are true (or not) in cases. For the rest of this section, we'll take cases in the sense of conceivable scenario, i.e., in the sense in which we used them to define conceptual validity. The points we made about different kinds of validity can be made about our new notions along similar lines: if we use a different idea of what counts as a "case" we will get different notions. And as logicians we will, eventually, consider a more permissive definition of case than we do here.

2.1 Joint possibility

Consider these two sentences:

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

Logic alone cannot tell us which, if either, of these sentences is true. Yet we can say that *if* the first sentence (B1) is true, *then* the second sentence (B2) must be false. Similarly, if B2 is true, then B1 must be false. There is no possible scenario where both sentences are true together. These sentences are incompatible with each other, they cannot all be true at the same time. This motivates the following definition:

> Sentences are JOINTLY POSSIBLE if and only if there is a case where they are all true together.

B1 and B2 are *jointly impossible*, while, say, the following two sentences are jointly possible:

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

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

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

G1 and G4 together entail that there are at least four martian giraffes at the park. This conflicts with G3, which implies that there are no more than two martian giraffes there. So the sentences G1–G4 are jointly impossible. They cannot all be true together. (Note that the sentences G1, G3 and G4 are jointly impossible. But if sentences are already jointly impossible, adding an extra sentence to the mix cannot make them jointly possible!)

2.2 Necessary truths, necessary falsehoods, and contingency

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

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

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

Sentence 2 is different. You do not need to look outside to know that it is true. Regardless of what the weather is like, it is either raining or it is not. That is a NECESSARY TRUTH.

Equally, you do not need to check the weather to determine whether or not sentence 3 is true. It must be false, simply as a matter of logic. It might be raining here and not raining across town; it might be raining now but stop raining even as you finish this sentence; but it is impossible for it to be both raining and not raining in the same place and at the same time. So, whatever the world is like, it is not both raining here and not raining here. It is a NECESSARY FALSEHOOD.

Something might *always* be true and still be contingent. For instance, if there never were a time when the universe contained fewer than seven things, then the sentence 'At least seven things exist' would always be true. Yet the sentence is contingent: the world could have been much, much smaller than it is, and then the sentence would have been false.

2.3 Necessary equivalence

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

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

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

Summary of logical notions

- An argument is VALID if there is no case where the premises are all true and the conclusion is not; it is INVALID otherwise.
- A NECESSARY TRUTH is a sentence that is true in every case.
- A NECESSARY FALSEHOOD is a sentence that is false in every case.
- A CONTINGENT SENTENCE is a sentence that is neither a necessary truth nor a necessary falsehood; a sentence that is true in some case and false in some other case.
- Two sentences are NECESSARILY EQUIVALENT if, in every case, they are both true or both false.
- A collection of sentences is JOINTLY POSSIBLE if there is a case where they are all true together; it is JOINTLY IMPOSSIBLE otherwise.

PART II

Truth-functional logic

CHAPTER 3

First steps to symbolization

3.1 Validity in virtue of form

Consider this argument:

- It is raining outside.
- If it is raining outside, then Jenny is miserable.
- Therefore: 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.
- Therefore: Dipan is an avid reader of Tolstoy.

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

- A
- If A, then B
- Therefore: B

This looks like an excellent argument *structure*. Indeed, surely any argument with this *structure* will be valid.

What about:

- Jenny is miserable.
- If it is raining outside, then Jenny is miserable.
- Therefore: It is raining outside.

The form of this argument is:

- *B*
- If *A* then *B*
- Therefore: *A*

Arguments of this form are generally invalid.

Be careful, though, not every argument of this form is sure to be invalid. It's possible to have an argument of this form that's valid – see if you can work out how! But most arguments of this form are invalid.

Consider an argument like:

- Jenny is happy or sad.
- Jenny is not happy.
- Therefore: Jenny is sad.

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

- A or B
- not-A
- Therefore: B

A superb structure! Here is another example:

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

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

- not-(A and B)
- A
- Therefore: not-B

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

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

It will have sentences like

$$(A \,\&\, (B \to \neg C))$$

3.2 Validity for special reasons

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

- Juanita is a vixen
- Therefore: Juanita is a fox

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

- Juanita is a vixen
- Therefore: Juanita is a cathedral

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

- The sculpture is green all over.
- Therefore: The sculpture is not red all over.

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

- The sculpture is green all over.
- Therefore: The sculpture is not shiny all over.

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

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

3.3 Atomic sentences

We started isolating the form of an argument, in §3.1, by replacing *subsentences* of sentences with individual letters. Thus in the first example of this section, 'it is raining outside' is a subsentence of 'If it is raining outside, then Jenny is miserable', and we replaced this subsentence with 'A'.

Our artificial language, TFL, pursues this idea absolutely ruthlessly. We start with some *atomic sentences*. These will be the basic building blocks out of which more complex sentences are built. We will use uppercase Roman letters for atomic sentences of TFL (except for X, Y, and Z which we reserve for metavariables). There are only twenty-three letters A–W, but there is no limit to the number of atomic sentences that we might want to consider. By adding subscripts to letters, we obtain new atomic sentences. So, here are five different atomic sentences of TFL:

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

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

A: It is raining outside
C: Jenny is miserable

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

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

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

CHAPTER 4

Connectives

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

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

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

4.1 Negation

Consider how we might symbolize these sentences:

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

3. Mary is not in Barcelona.

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

> B: Mary is in Barcelona.

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

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

> If a sentence can be paraphrased as 'it is not the case that X' it can be symbolised as $\neg X$.

It will help to offer a few more examples:

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

Let us use the following representation key:

> R: The widget is replaceable

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

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

But some care is needed when handling negations. Consider:

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

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

4.2 Conjunction

Consider these sentences:

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

We will need separate atomic sentences of TFL to symbolize sentences 9 and 10; perhaps

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

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

> If a sentence can be paraphrased as '$\mathcal{X}$ and $\mathcal{Y}$'
> it can be symbolised as $\mathcal{X} \,\&\, \mathcal{Y}$.

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

Some more examples will bring out this point:

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

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

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

Sentence 14 is slightly more complicated. The word 'although' sets up a contrast between the first part of the sentence and the second part. Nevertheless, the sentence tells us both that Barbara is energetic and that she is not athletic. In order to make each of the conjuncts an atomic sentence, we need to replace 'she' with 'Barbara'. So we can paraphrase sentence 14 as, '*Both* Barbara is energetic, *and* Barbara is not athletic'. The second conjunct contains a negation, so we paraphrase further: '*Both* Barbara is energetic *and it is not the case that* Barbara is athletic'. Now we can symbolize this with the TFL sentence $(E \,\&\, \neg B)$. Note that we have lost all sorts of nuance in this symbolization. There is a distinct difference in tone between sentence 14 and 'Both Barbara is energetic and it is not the case that Barbara is athletic'. TFL does not (and cannot) preserve these nuances.

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

We can add these to our toolbox for symbolisation:

If a sentence can be paraphrased as

- ‘X and Y’
- ‘X but Y’
- ‘Both X and Y’
- ‘Although X, Y’

it can be symbolised as $X \& Y$.

4.3 Disjunction

Consider these sentences:

16. Fatima will play videogames, or she will watch movies.
17. Fatima or Omar will play videogames.

For these sentences we can use this symbolization key:

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

However, we will again need to introduce a new symbol. Sentence 16 is symbolized by $(F \vee M)$. The connective is called DISJUNCTION. We also say that F and M are the DISJUNCTS of the disjunction $(F \vee M)$.

Sentence 17 is only slightly more complicated. There are two subjects, but the English sentence only gives the verb once. However, we can paraphrase sentence 17 as ‘Fatima will play videogames, or Omar will play videogames’. Now we can obviously symbolize it by $(F \vee O)$ again.

If a sentence can be paraphrased as ‘X or Y’
it can be symbolised as $X \vee Y$.

X and Y need to be the sorts of things that can be whole sentences. We cannot simply take X to be ‘Fatima’, instead we first need to paraphrase it as in 16

Sometimes in English, the word ‘or’ is used in a way that excludes the possibility that both disjuncts are true. This is called an EXCLUSIVE

OR. An *exclusive or* is clearly intended when it says, on a restaurant menu, 'Entrees come with either soup or salad': you may have soup; you may have salad; but, if you want *both* soup *and* salad, then you have to pay extra.

At other times, the word 'or' allows for the possibility that both disjuncts might be true. This is probably the case with sentence 17, above. Fatima might play videogames alone, Omar might play videogames alone, or they might both play. Sentence 17 merely says that *at least* one of them plays videogames. This is called an INCLUSIVE OR. The TFL symbol '$\lor$' always symbolizes an *inclusive or*.

TFL can symbolise the exclusive or, it just doesn't do it with '$\lor$'. We will discuss this in §??

4.4 Conditional

Consider the sentence

18. If Jean is in Paris, then she is in France.

Let's use the following symbolization key:

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

Sentence 18 is roughly of this form: 'if P, then F'. We will use the symbol '$\rightarrow$' to symbolize this 'if..., then...' structure. So we symbolize sentence 18 by $(P \rightarrow F)$. The connective '$\rightarrow$' is called THE CONDITIONAL. Here, P is called the ANTECEDENT of the conditional $(P \rightarrow F)$, and F is called the CONSEQUENT.

> If a sentence can be paraphrased as
> 'If X, then Y' it can be symbolised as $X \rightarrow Y$.

Now consider the sentences

19. If Jean is in Paris, she's in France.
20. Jean is in France if she is in Paris.

Sentence 19 and 20 are just rephrasings of 18. So we will again symbolise them as $(P \rightarrow F)$.

Now consider

21. Jean is in France only if she is in Paris.

21 is also a conditional. In 18–20 we took as an antecedent the part of sentence that has the word 'if' in it. It might then be tempting to do the same here and symbolize this as $(P \to F)$. That would be a mistake. Your knowledge of geography tells you that sentence 18 is unproblematically true: there is no way for Jean to be in Paris that doesn't involve Jean being in France. But sentence 21 is not so straightforward: were Jean in Dieppe, Lyons, or Toulouse, Jean would be in France without being in Paris, thereby rendering sentence 21 false. Since geography alone dictates the truth of sentence 18, whereas travel plans (say) are needed to know the truth of sentence 21, they must mean different things. In fact, sentence 21 can be paraphrased as 'If Jean is in France, then Jean is in Paris'. So we can symbolize it by $(F \to P)$: the other way around to 18.

If a sentence can be paraphrased as

- 'If X, then Y'
- 'If X, Y'
- 'Y if X'
- 'X only if Y'

it can be symbolised as $X \to Y$.

It is important to bear in mind that the connective '$\to$' tells us only that, if the antecedent is true, then the consequent is true. It says nothing about a *causal* connection between two events (for example). In fact, we lose a huge amount when we use $\to$ to symbolize English conditionals. We will return to this in §§10.1 and 15.

4.5 Biconditional

Consider these sentences:

22. Laika is a dog only if she is a mammal
23. Laika is a dog if she is a mammal
24. Laika is a dog if and only if she is a mammal

We will use the following symbolization key:

D: Laika is a dog

M: Laika is a mammal

Sentence 22, for reasons discussed above, can be symbolized by $D \rightarrow M$.

Sentence 23 is importantly different. It can be paraphrased as, 'If Laika is a mammal then Laika is a dog'. So it can be symbolized by $M \rightarrow D$.

Sentence 24 says something stronger than either 22 or 23. It can be paraphrased as 'Laika is a dog if Laika is a mammal, and Laika is a dog only if Laika is a mammal'. This is just the conjunction of sentences 22 and 23. So we can symbolize it as $((D \rightarrow M) \,\&\, (M \rightarrow D))$. We call this a BICONDITIONAL, because it entails the conditional in both directions.

We could treat every biconditional this way. So, just as we do not need a new TFL symbol to deal with *exclusive or*, we do not really need a new TFL symbol to deal with biconditionals. Because the biconditional occurs so often, however, we will use the symbol '$\leftrightarrow$' for it. We can then symbolize sentence 24 with the TFL sentence $(D \leftrightarrow M)$.

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

> If a sentence can be paraphrased as 'X if and only if Y' it can be symbolised as $X \leftrightarrow Y$.

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

CHAPTER 5

Sentences of TFL

We have seen connectives in TFL, but there is another part of the language of TFL: brackets. These provide TFL with its grammar.

Sentences of TFL will be things like:

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

$$\neg(A \,\&\, B)$$

There are two purposes of grammar.

The first is to avoid nonsense. Just as in English we want to avoid

The and dog brown or is.

in TFL we want to avoid

$$A \,\&\, \vee B \rightarrow$$

which would be nonsense.

The second purpose is to avoid ambiguity. Just as we wish to avoid

John's tired and Sue's tall or Rob's short

in English, in TFL we wish to avoid ambiguity. Brackets are sort of like punctuation in English, they help us know what goes with what.

1. John's tired and Sue's tall or Rob's short

By adding commas this can either be read as:

2. John's tired, and Sue's tall or Rob's short.
3. John's tired and Sue's tall, or Rob's short

In TFL we follow mathematics in using brackets to do this.

4. John's tired and Sue's tall or Rob's short

By adding commas this can either be read as:

5. John's tired, and Sue's tall or Rob's short.
6. John's tired and Sue's tall, or Rob's short

In mathematics,

7. $9 + 3 \times 4$

can either be read as:

8. $9 + (3 \times 4)$ $\quad (= 9 + 12 = 21)$
9. $(9 + 3) \times 4$ $\quad (= 12 \times 4 = 48)$

In TFL,

10. $A \mathbin{\&} B \vee C$

can either be read as:

11. $A \mathbin{\&} (B \vee C)$
12. $(A \mathbin{\&} B) \vee C$

In fact, we will say that $A \mathbin{\&} B \vee C$ is not a sentence of TFL at all. It is only expressions which have proper grammar which count as sentences.

To make this precise, this chapter offers a formal definition of what it is to be a sentence in TFL.

5.1 Syntactic rules of TFL

We give rules for what counts as a sentence which will together give us a definition.

We start with the rule:

1. Any uppercase Roman letters A–W, or with subscripts, e.g., $A_1, B_3, A_{100}, J_{375}$, are sentences of TFL.
   These are called atomic sentences.

We only permit use of A–W because X, Y, and Z are reserved for metavariables.

Our second rule says:

2. If X is a sentence of TFL, then so is $\neg X$.

By rule 1, we know that A is a sentence. Rule 2 then allows us to conclude that $\neg A$ is also a sentence. We could then apply it again and conclude that $\neg\neg A$ is also a sentence.

FORMATION TREES help us keep track of this process. For the case of $\neg\neg A$ this would be:

$$\begin{array}{c} \neg\,\neg A \\ | \\ \neg\, A \\ | \\ A \end{array}$$

Our third rule says:

3. If X and Y are sentences, then so is $(X \,\&\, Y)$.

By rule 1, B_1 and D are both sentences. So rule 3 allows us to conclude that $(B_1 \,\&\, D)$ is a sentence. We might then apply rule 2 to conclude that $\neg(B_1 \,\&\, D)$ is also a sentence.

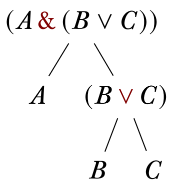
$$\begin{array}{c} \neg\ (B_1 \,\&\, D) \\ | \\ (B_1 \,\&\, D) \\ / \quad \backslash \\ B_1 \qquad D \end{array}$$

We then give similar rules for each of our other connectives: $\vee, \rightarrow$ and $\leftrightarrow$.

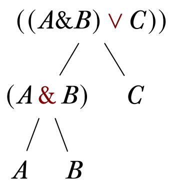
We summarise this in a definition:

1. Every atomic sentence is a sentence.
2. If X is a sentence, then $\neg X$ is a sentence.
3. If X and Y are sentences, then $(X \,\&\, Y)$ is a sentence.
4. If X and Y are sentences, then $(X \vee Y)$ is a sentence.
5. If X and Y are sentences, then $(X \rightarrow Y)$ is a sentence.
6. If X and Y are sentences, then $(X \leftrightarrow Y)$ is a sentence.
7. Nothing else is a sentence.

For example, consider $(A \,\&\, (B \vee C))$ we can check this is a sentence by drawing the following formation tree:



Each of the steps here tracks one of the rules of what it is to be a sentence. So we can conclude that this is a sentence of TFL. This also helps us see how to read it. It has a different formation tree from $((A \,\&\, B) \vee C)$:



$A \,\&\, B \vee C$ is not a sentence of TFL. It cannot be broken up into smaller sentences by any of our rules. It needs brackets to be a well-formed sentence. The different formations will be important when we describe truth-tables for these sentences. $((A \,\&\, B) \vee C)$ and $((A \,\&\, B) \vee C)$ will differ in when they are true.

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

- My parents are ancestors of mine.

and then offer further clauses like:

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

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

One more example: consider $\neg(P \,\&\, \neg(\neg Q \vee P))$ we can check this is a sentence by drawing the following formation tree:

$$\neg\ (P \,\&\, \neg(\neg Q \vee P))$$

$$|$$

$$(P\ \&\ \neg(\neg Q \vee P))$$

$$/ \qquad \backslash$$

$$P \qquad \neg\ (\neg Q \vee P)$$

$$/ \quad \backslash$$

$$\neg\ Q \qquad P$$

$$|$$

$$Q$$

each of the steps here tracks one of the rules of what it is to be a sentence. So we can conclude that this is a sentence of TFL. This also helps us see how to read it. The will be important when we consider the circumstances under which a particular sentence would be true or false. The sentence $\neg Q$ is true if and only if the sentence Q is false, and so on through the structure of the sentence, until we arrive at the atomic components. We will return to this point in Part III.

This displays clearly the recursive structure of the tree. The sentences further up the tree are formed by one of the formation rules from the sentences further down the tree. The main connective of each sentence is in red.

When drawing these trees we have highlighted a particular connective on each of our nodes. We call that connective the MAIN CONNECTIVE of the sentence.

> The MAIN CONNECTIVE of sentence is the last connective that was introduced in the construction of the sentence.

In the case of $((\neg E \vee F) \rightarrow \neg\neg G)$, the main connective is $\rightarrow$. Here we can see that the whole sentence can be described in the form $(X \rightarrow Y)$ with both X and Y being complete sentences (put $X = (\neg E \vee F)$ and $Y = \neg\neg G)$. That's enough to see that $\rightarrow$ is the main operator. In the case of $\neg\neg\neg D$, the main logical operator is the very first $\neg$ sign. This is because we can see the sentence as having the form $\neg X$ with X being the complete sentence $\neg\neg D$. In the case of $(P \,\&\, \neg(\neg Q \vee R))$, the main logical operator is &: it's an $(X \,\&\, Y)$ with X as P and Y as $\neg(\neg Q \vee R)$.

We also say:

> The SCOPE of an instance of a connective is the subsentence for which it is the main connective.

For example, in:

$$(P \,\&\, (\neg(R \,\&\, Q) \leftrightarrow P))$$

The scope of the $\neg$ is $\neg(R \,\&\, Q)$.

To see this, we can draw the formation tree:

$$(P \,\&\, (\neg(R \,\&\, Q) \leftrightarrow P))$$

$$P \qquad (\neg(R \,\&\, Q) \leftrightarrow P))$$

$$\neg\,(R \,\&\, Q) \qquad P$$

$$(R \,\&\, Q)$$

$$R \qquad Q$$

$\neg(R \& Q)$ is the subsentence in which $\neg$ is the main logical operator. We have worked that out by drawing the formation tree and finding the lowest sentence in which that connective instance appears.

We might informally describe the scope as the part of the whole sentence that that connective cares about. For the purposes of the connective & in $(A \vee (B \& \neg C))$, the truth value of A is irrelevant. It is only what is going on with B and $\neg C$ that matters for it. The notion of scope will be very important when we look at First Order Logic.

5.2 Bracketing conventions

Strictly speaking, $A \& B$ is not a sentence of TFL. When we introduce a connective $\&, \vee, \rightarrow$ or $\leftrightarrow$, strictly speaking, we must include brackets. It is only $(A \& B)$ The reason for this rule is that we might use $(A \& B)$ as a subsentence in a more complicated sentence. For example, we might want to negate $(A \& B)$, obtaining $\neg(A \& B)$. If we just had $A \& B$ without the brackets and put a negation in front of it, we would have $\neg A \& B$. It is most natural to read this as meaning the same thing as $(\neg A \& B)$, but this may be very different from $\neg(A \& B)$.

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

First, we allow ourselves to omit the *outermost* brackets of a sentence. Thus we allow ourselves to write $A \& B$ instead of the sentence $(A \& B)$. However, we must remember to put the brackets back in, when we want to embed the sentence into a more complicated sentence!

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

Combining these two conventions, we can rewrite the unwieldy sentence

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

rather more clearly as follows:

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

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

CHAPTER 6

Symbolising complex sentences

In §4 we discussed how to symbolise sentences. But we mostly focused on simple sentences. Sentences of English, though, might end up having much more complex structure.

Our general strategy will be:

1. See if the sentence can be reformulated naturally with 'if', 'and' etc between sentences. If not, use an atomic sentence.
2. Replace the 'and' with &, or as appropriate (with brackets).
3. Repeat the procedure with the sentences connected by &.

We can now see some cases where the brackets are very important.

Consider, for example, how negation might interact with conjunction. Consider:

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

Sentence 1 can be paraphrased as 'It is not the case that: both you will get soup and you will get salad'. Using this symbolization key:

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

We would symbolize 'both you will get soup and you will get salad' as $(S_1 \& S_2)$. To symbolize sentence 1, then, we simply negate the whole sentence, thus: $\neg(S_1 \& S_2)$.

Sentence 2 is a conjunction: you *will not* get soup, and you *will* get salad. 'You will not get soup' is symbolized by $\neg S_1$. So to symbolize sentence 2 itself, we offer $(\neg S_1 \& S_2)$.

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

3. You will not have soup, or you will not have salad.
4. You will have neither soup nor salad.
5. You get either soup or salad (but not both).

Using the same symbolization key as before, sentence 3 can be paraphrased in this way: 'Either *it is not the case that* you get soup, or *it is not the case that* you get salad'. To symbolize this in TFL, we need both disjunction and negation. 'It is not the case that you get soup' is symbolized by $\neg S_1$. 'It is not the case that you get salad' is symbolized by $\neg S_2$. So sentence 3 itself is symbolized by $(\neg S_1 \vee \neg S_2)$.

Sentence 4 also requires negation. It can be paraphrased as, '*It is not the case that* either you get soup or you get salad'. Since this negates the entire disjunction, we symbolize sentence 4 with $\neg(S_1 \vee S_2)$.

Sentence 5 is an *exclusive or*. We can break the sentence into two parts. The first part says that you get one or the other. We symbolize this as $(S_1 \vee S_2)$. The second part says that you do not get both. We can paraphrase this as: 'It is not the case both that you get soup and that you get salad'. Using both negation and conjunction, we symbolize this with $\neg(S_1 \& S_2)$. Now we just need to put the two parts together. As we saw above, 'but' can usually be symbolized with &. Sentence 5 can thus be symbolized as $((S_1 \vee S_2) \& \neg(S_1 \& S_2))$.

Sometimes in English it is ambiguous whether the 'or' is to be understood in English inclusively or exclusively. When symbolising into TFL, one needs to decide if you want to add 'but not both' or not.

It is also important to note that even though the TFL symbol '$\vee$' always symbolizes *inclusive or*, we can symbolize an *exclusive or* in TFL. We just have to use a few of our other symbols as well.

- If a sentence can be paraphrased as 'neither X or Y' it can be symbolised as $\neg(X \vee Y)$.
- If a sentence paraphrased as 'either X or Y, but not both' (an *exclusive or*) it can be symbolised as $((X \vee Y) \,\&\, \neg(X \,\&\, Y))$.

6.1 Unless

An especially difficult case is when we use the English-language connective 'unless':

6. Unless you wear a jacket, you will catch a cold.
7. You will catch a cold unless you wear a jacket.

These two sentences are equivalent. They are also equivalent to the following:

8. If you do not wear a jacket, then you will catch a cold.
9. If you do not catch a cold, then you wore a jacket.
10. Either you will wear a jacket or you will catch a cold.

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

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

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

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

If a sentence can be paraphrased as 'Unless X, Y', then it can be symbolized as $(X \vee Y)$.

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

CHAPTER 7

Ambiguity

In English, sentences can be AMBIGUOUS, i.e., they can have more than one meaning. There are many sources of ambiguity. One is *lexical ambiguity:* a sentence can contain words which have more than one meaning. For instance, 'bank' can mean the bank of a river, or a financial institution. So I might say that 'I went to the bank' when I took a stroll along the river, or when I went to deposit a check. Depending on the situation, a different meaning of 'bank' is intended, and so the sentence, when uttered in these different contexts, expresses different meanings.

A different kind of ambiguity is *structural ambiguity.* This arises when a sentence can be interpreted in different ways, and depending on the interpretation, a different meaning is selected. A famous example due to Noam Chomsky is the following:

- Flying planes can be dangerous.

There is one reading in which 'flying' is used as an adjective which modifies 'planes'. In this sense, what's claimed to be dangerous are airplanes which are in the process of flying. In another reading, 'flying' is a gerund: what's claimed to be dangerous is the act of flying a plane. In the first case, you might use the sentence to warn someone who's about to launch a hot air baloon. In the second case, you might use it to counsel someone against becoming a pilot.

When the sentence is uttered, usually only one meaning is intended. Which of the possible meanings an utterance of a sentence intends is determined by context, or sometimes by how it is uttered (which parts of the sentence are stressed, for instance). Often one interpretation is much more likely to be intended, and in that case it will even be difficult to “see” the unintended reading. This is often the reason why a joke works, as in this example from Groucho Marx:

- One morning I shot an elephant in my pajamas.
- How he got in my pajamas, I don’t know.

Ambiguity is related to, but not the same as, vagueness. An adjective, for instance ‘rich’ or ‘tall,’ is VAGUE when it is not always possible to determine if it applies or not. For instance, a person who’s 6 ft 4 in (1.9 m) tall is pretty clearly tall, but a building that size is tiny. Here, context has a role to play in determining what the clear cases and clear non-cases are (‘tall for a person,’ ‘tall for a basketball player,’ ‘tall for a building’). Even when the context is clear, however, there will still be cases that fall in a middle range.

In TFL, we generally aim to avoid ambiguity. We will try to give our symbolization keys in such a way that they do not use ambiguous words or disambiguate them if a word has different meanings. So, e.g., your symbolization key will need two different sentence letters for ‘Rebecca went to the (money) bank’ and ‘Rebecca went to the (river) bank.’ Vagueness is harder to avoid. Since we have stipulated that every case (and later, every valuation) must make every basic sentence (or sentence letter) either true or false and nothing in between, we cannot accommodate borderline cases in TFL.

It is an important feature of sentences of TFL that they *cannot* be structurally ambiguous. Every sentence of TFL can be read in one, and only one, way. This feature of TFL is also a strength. If an English sentence is ambiguous, TFL can help us make clear what the different meanings are. Although we are pretty good at dealing with ambiguity in everyday conversation, avoiding it can sometimes be terribly important. Logic can then be usefully applied: it helps philosopher express their thoughts clearly, mathematicians to state their theorems rigorously, and software engineers to specify loop conditions, database queries, or verification criteria unambiguously.

Stating things without ambiguity is of crucial importance in the law as well. Here, ambiguity can, without exaggeration, be a matter of life and death. Here is a famous example of where a death sentence hinged on the interpretation of an ambiguity in the law. Roger Casement (1864–1916) was a British diplomat who was famous in his time for publicizing human-rights violations in the Congo and Peru (for which he was knighted in 1911). He was also an Irish nationalist. In 1914–16, Casement secretly travelled to Germany, with which Britain was at war at the time, and tried to recruit Irish prisoners of war to fight against Britain and for Irish independence. Upon his return to Ireland, he was captured by the British and tried for high treason.

The law under which Casement was tried is the *Treason Act of 1351*. That act specifies what counts as treason, and so the prosecution had to establish at trial that Casement's actions met the criteria set forth in the Treason Act. The relevant passage stipulated that someone is guilty of treason

> if a man is adherent to the King's enemies in his realm, giving to them aid and comfort in the realm, or elsewhere.

Casement's defense hinged on the last comma in this sentence, which is not present in the original French text of the law from 1351. It was not under dispute that Casement had been 'adherent to the King's enemies', but the question was whether being adherent to the King's enemies constituted treason only when it was done in the realm, or also when it was done abroad. The defense argued that the law was ambiguous. The claimed ambiguity hinged on whether 'or elsewhere' attaches only to 'giving aid and comfort to the King's enemies' (the natural reading without the comma), or to both 'being adherent to the King's enemies' and 'giving aid and comfort to the King's enemies' (the natural reading with the comma). Although the former interpretation might seem far fetched, the argument in its favor was actually not unpersuasive. Nevertheless, the court decided that the passage should be read with the comma, so Casement's antics in Germany were treasonous, and he was sentenced to death. Casement himself wrote that he was 'hanged by a comma'.

We can use TFL to symbolize both readings of the passage, and thus to provide a disambiguiation. First, we need a symbolization key:

> A: Casement was adherent to the King's enemies in the realm.
> G: Casement gave aid and comfort to the King's enemies in the realm.
> B: Casement was adherent to the King's enemies abroad.
> H: Casement gave aid and comfort to the King's enemies abroad.

The interpretation according to which Casement's behavior was not treasonous is this:

- $A \vee (G \vee H)$

The interpretation which got him executed, on the other hand, can be symbolized by:

- $(A \lor B) \lor (G \lor H)$

Remember that in the case we're dealing with Casement, was adherent to the King's enemies abroad (B is true), but not in the realm, and he did not give the King's enemies aid or comfort in or outside the realm (A, G, and H are false).

One common source of structural ambiguity in English arises from its lack of parentheses. For instance, if I say 'I like movies that are not long and boring', you will most likely think that what I dislike are movies that are long and boring. A less likely, but possible, interpretation is that I like movies that are both (a) not long and (b) boring. The first reading is more likely because who likes boring movies? But what about 'I like dishes that are not sweet and flavorful'? Here, the more likely interpretation is that I like savory, flavorful dishes. (Of course, I could have said that better, e.g., 'I like dishes that are not sweet, yet flavorful'.) Similar ambiguities result from the interaction of 'and' with 'or'. For instance, suppose I ask you to send me a picture of a small and dangerous or stealthy animal. Would a leopard count? It's stealthy, but not small. So it depends whether I'm looking for small animals that are dangerous or stealthy (leopard doesn't count), or whether I'm after either a small, dangerous animal or a stealthy animal (of any size).

These kinds of ambiguities are called *scope ambiguities*, since they depend on whether or not a connective is in the scope of another. For instance, the sentence, '*Avengers: Endgame* is not long and boring' is ambiguous between:

1. *Avengers: Endgame* is not: both long and boring.
2. *Avengers: Endgame* is both: not long and boring.

Sentence 2 is certainly false, since *Avengers: Endgame* is over three hours long. Whether you think 1 is true depends on if you think it is boring or not. We can use the symbolization key:

B: *Avengers: Endgame* is boring.
L: *Avengers: Endgame* is long.

Sentence 1 can now be symbolized as '$\lnot(L \,\&\, B)$', whereas sentence 2 would be '$\lnot L \,\&\, B$'. In the first case, the '&' is in the scope of '$\lnot$', in the second case '$\lnot$' is in the scope of '&'.

The sentence 'Tai Lung is small and dangerous or stealthy' is ambiguous between:

3. Tai Lung is either both small and dangerous or stealthy.
4. Tai Lung is both small and either dangerous or stealthy.

We can use the following symbolization key:

D: Tai Lung is dangerous.
S: Tai Lung is small.
T: Tai Lung is stealthy.

The symbolization of sentence 3 is '$(S \& D) \vee T$' and that of sentence 4 is '$S \& (D \vee T)$'. In the first, &is in the scope of $\vee$, and in the second $\vee$is in the scope of &.

CHAPTER 8

Use and mention

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

8.1 Quotation conventions

Consider these two sentences:

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

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

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

- 'Justin Trudeau' is the Prime Minister.

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

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

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

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

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

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

8.2 Quotation in Logic

Since we are introducing a formal language, we never really *use* the language in the way we would use English or German; we are always mentioning it. And it's clear from the way it's written that it's not part of English, so logicians often drop the quotation marks around logic. Really we should say

- 'D' is an atomic sentence of TFL.
- '$(D \,\&\, (B \vee C))$' is a sentence of TFL.

But we don't generally bother, and will typically just write

- D is an atomic sentence of TFL.
- $(D \,\&\, (B \vee C))$ is a sentence of TFL.

This textbook follows this convention.

8.3 Metavariables

Often we do not just want to talk about *specific* expressions of TFL. We also want to be able to talk about *any arbitrary* sentence of TFL. Indeed, we had to do this in §5, when we presented the recursive definition of a sentence of TFL. We used letters from the end of the alphabet to do this, namely:

$$X, Y, Z, X_1, Y_{100}, \ldots$$

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

2. If X is a sentence, then $\neg X$ 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, it would only tell us about the specific case for A. We don't want to say something about A specifically, but about all sentences. For example this should also give us that

- If $(A \vee (B \,\&\, C))$ is a sentence, then $\neg\,(A \vee (B \,\&\, C))$ is a sentence.

We get this by using a metavariable X. It stands in place of an arbitrary expression.

To emphasize, then:

X (also Y, Z) is a symbol (called a METAVARIABLE) in augmented English, which we use to talk about any expression of our formal language. A (also $B, C, \ldots$) is a particular atomic sentence of TFL.

PART III

Truth tables

CHAPTER 9

Truth rules for the connectives of TFL

We now move to looking at when sentences of TFL are true or false. We will give precise rules for determining this. The important feature of truth functional logic is that the truth value of a complex sentence, such as $A \vee (B \& C)$ is determined just by the truths of is component parts, that is A, B and C. If we're told whether A, B and C are true or false, then we will be able to say whether $A \vee (B \& C)$ is true or false.

To be able to do this, we need to describe how the truth values are to be combined. We work through each of our connectives describing the rules governing it.

9.1 Negation

Consider:

1. Bristol is not in France.
2. Bristol is not in England.

'Bristol is in France' is false, so 'Bristol is not in France' is true. 'Bristol is in England' is true, so 'Bristol is not in England' is false.

In general, to know whether a sentence of the form $\neg X$ is true. This depends on whether X is true or not in the way:

- ▹ If X is true, then $\neg X$ is false.
- ▹ If X is false, then $\neg X$ is true.

We record this in shorthand:

If X is:		then $\neg X$ is:
T	$\rightsquigarrow$	F
F	$\rightsquigarrow$	T

We have abbreviated 'True' with 'T' and 'False' with 'F'. (But just to be clear, the two truth values are True and False; the truth values are not *letters*!)

9.2 Conjunction

Recall that $X \& Y$ was used to symbolise 'X and Y'.

Consider:

3. She can speak German and French.

If she can speak German and she can speak French, then this is true, but otherwise it is false.

More generally, the rule governing & is:

- ▹ If X and Y are both true, then $X \& Y$ is true.
- ▹ Otherwise, $X \& Y$ is false.

Which we summarise

If X is:	and Y is:		then $X \& Y$ is:
T	T	$\rightsquigarrow$	T
T	F	$\rightsquigarrow$	F
F	T	$\rightsquigarrow$	F
F	F	$\rightsquigarrow$	F

Note that conjunction is *symmetrical*. The truth value for $X \& Y$ is always the same as the truth value for $Y \& X$.

9.3 Disjunction

Disjunction is a bit more subtle. Consider:

4. She can speak German or French.

If she cannot speak either German or French, then this is false. If she can speak German but not French, then it is true, and if she can speak French but not German it is also true. We have the general rules:

- ▹ If X and Y are both false, then $X \vee Y$ is false.
- ▹ If X is true and Y are false, then $X \vee Y$ is true.
- ▹ If X is false and Y are true, then $X \vee Y$ is true.

But what if she can speak both? Is it true or false? We have already pointed out that in English there are two kinds of disjunctions: an *inclusive* and an *exclusive* one.

For the inclusive or, we might whisper a "or both" after it; whereas for the exclusive or, we'd want to whisper a "but not both":

5. She speaks German or French (or both).
6. She speaks German or French (but not both).

In logic there can be no ambiguity. We choose that $\vee$ stands for the *inclusive or*. That is, we give the final rule:

- ▹ If X and Y are both true, then $X \vee Y$ is true.

So, when doing symbolisations, one should only symbolise a sentence as $X \vee Y$ if it is to be read as the *inclusive or*. To symbolise the exclusive or, you need to use the more complex sentence: $(X \vee Y) \,\&\, \neg(X \,\&\, Y)$, which essentially makes explicit the whispered "but not both". Sometimes the English is ambiguous, in which case one should point that out when symbolising and give the two alternative symbolisations.

Consider:

7. She either ate pizza or pasta.

Maybe this is an exclusive or, though an alternative treatment is to take this to be an inclusive or, but note that there's a implicit, or missing, premise: that she didn't eat both.

To summarise the rules for $\vee$:

If X is:	and Y is:		then $X \vee Y$ is:
T	T	$\rightsquigarrow$	T
T	F	$\rightsquigarrow$	T
F	T	$\rightsquigarrow$	T
F	F	$\rightsquigarrow$	F

Like conjunction, disjunction is symmetrical.

9.4 Conditional

Suppose you are a bartender considering the conditional

8. If she is drinking a beer, then she is over eighteen.

Here is what we'll say about this conditional:

- ▹ If she's drinking beer and is sixteen, then it is false. (You should kick her out of the bar.)
- ▹ If she is drinking beer and is nineteen, then it's true.
- ▹ If she is drinking coke, then it's true. (No need to check her age)

We summarise these rules:

If X is:	and Y is:		then $X \to Y$ is:
T	T	$\rightsquigarrow$	T
T	F	$\rightsquigarrow$	F
F	T	$\rightsquigarrow$	T
F	F	$\rightsquigarrow$	T

In this case, it's very important to remember which way around it goes. The TF-line is different to the FT-line.

This is why the terms 'antecedent' and 'consequent' are so useful.

In $X \to Y$, X is called the ANTECEDENT, and Y the CONSEQUENT.

We can redescribe this rule: If the antecedent is true and the consequent false, then the conditional sentence is false, otherwise it is true.

The TFL connective $\to$ is *stipulated* to be governed by these rules. It is sometimes called the MATERIAL CONDITIONAL to highlight that it is governed by these rules.

Subjunctive conditionals

Some English sentences that have the form 'if... then...' do not fit with these rules. Consider

9. If the UK had voted to remain in the EU, then the moon would be made of cheese.

This has a false antecedent (we voted to exit), and a false consequent (I certainly can't fly). But the whole sentence would be judged to be true. So this doesn't fit with the rule we've given for $\to$.

There are two kinds of sentences of the form 'if..., then...' in English: indicative and subjunctive ones. Consider the two sentences:

10. If Oswald didn't kill Kennedy, someone else did.
11. If Oswald hadn't killed Kenney, someone else would have.

The former uses the INDICATIVE CONDITIONAL and is plausibly true. After all, we know that Kennedy was assassinated. The latter uses the SUBJUNCTIVE CONDITIONAL and is plausibly false. Unless there was a conspiracy plot, if Oswald hadn't have killed Kenney then Kennedy wouldn't have been assassinated. The term COUNTERFACTUAL CONDITIONAL is also often used.[1]

Whilst we use the English 'if...then...' for both sentences 10 and 11, they are very different kinds of conditionals. The conditional of TFL does a terrible job at symbolising the subjunctive conditional which aren't governed by these rules. In fact TFL just doesn't have the resources to consider subjunctive conditionals, they are not "truth functional". We will discuss this in the next chapter, but for now we simply not that if you have to symbolise a subjunctive conditional in TFL it is then a matter of judgement whether to do it as $X \to Y$, or to simply use an atomic sentence.

The material conditional does a much better job at symbolising the indicative conditionals, though it is a matter of substantial philosophical debate about exactly how well they do. We briefly mention some of the worries in §15.

[1]Strictly speaking, we can use subjunctive conditionals with true antecedents, whereas a counterfactual conditional has to have a false antecedent (to be "counter to the fact").

9.5 Biconditional

Consider

12. Sue is coming to the party if and only if Maya is coming.

If Sue and Maya are both coming, then this is true. If they're both not coming, then it is also true. If only one of them is coming, then it is false.

If X is:	and Y is:		then $X \leftrightarrow Y$ is:
T	T	$\rightsquigarrow$	T
T	F	$\rightsquigarrow$	F
F	T	$\rightsquigarrow$	F
F	F	$\rightsquigarrow$	T

You can see a summary of the rules for all the connectives, for ease of reference, in the 'Quick Reference Appendix' §3.2

CHAPTER 10

Truth-functional connectives

In this chapter, we reflect on truth-functional logic and the connectives we've used.

10.1 Non truth-functional connectives

Let's introduce an important idea.

> 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).

Every connective in TFL is truth-functional. We were able to give rules to determine what the truth value of a sentence $\neg X$ is depending only on the truth value of X. The truth value of X uniquely determines the truth value of $\neg X$. The same is true for all the other connectives of TFL ($\&, \vee, \rightarrow, \leftrightarrow$). This is what gives TFL its name: it is *truth-functional logic.*

This then means that to determine the truth value of any TFL sentence, we only need to know the truth value of the atomic sentences it includes. We will see exactly how to do this in §11.

In plenty of languages there are connectives that are not truth-functional. We here describe just a few:

Necessarily

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

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

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

Subjunctive conditionals

We said that $\rightarrow$ was pretty bad at capturing *subjunctive conditionals* of English. The problem is that a subjunctive conditional is not truth functional. Consider the two sentences:

3. If Mitt Romney had won the 2012 election, then he would have been the 45th President of the USA.
4. 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 3 is true; sentence 4 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.

$\rightarrow$ is the best that can be done at symbolising subjunctive conditionals of English in TFL. TFL just doesn't have the required resources as the subjunctive conditional is not truth functional.

10.2 Symbolizing versus translating

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

When we symbolize a sentence or an argument in TFL, we ignore everything *besides* the contribution that the truth values of a component might make to the truth value of the whole. There are subtleties to our ordinary claims that far outstrip their mere truth values. Sarcasm; poetry; snide implicature; emphasis; these are important parts of everyday discourse, but none of this is retained in TFL. As remarked in §4, TFL cannot capture the subtle differences between the following English sentences:

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

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

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

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

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

Other textbooks will understand this as a stipulation that the TFL sentence 'L' should *mean* that Dana is a logician, and that the TFL sentence 'N' should *mean* that Dana is a nice person, but TFL just is totally unequipped to deal with *meaning*. The preceding symbolization key is doing no more and no less than stipulating that the TFL sentence 'L' should take the same truth value as the English sentence 'Dana is a logician' (whatever that might be), and that the TFL sentence 'N' should take the same truth value as the English sentence 'Dana is a nice person' (whatever that might be).

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

CHAPTER 11

Complete truth tables

In §9 we described how the truth values of two sentences, X and Y, should combine to determine the truth of a sentence such as $X \vee Y$.

We will now describe how to extend this reasoning to determine the truths of more complex sentences such as $(\neg I \rightarrow H) \& H$.

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

> A VALUATION is any assignment of truth values to particular atomic sentences of TFL.

We describe how to fill out TRUTH TABLES. Each row of a truth table is a valuation. The entire truth table represents all possible valuations; thus the truth table provides us with a means to calculate the truth values of complex sentences, on each possible valuation. This is easiest to explain by example.

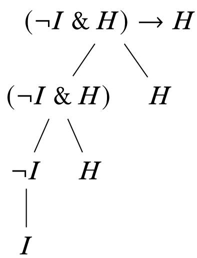
11.1 A worked example

Consider the sentence $(\neg I \,\&\, H) \to H$. We will give a *truth table* which lists all the valuations and says whether this sentence is true or false on each of them. The valuations assign either True or False to each atomic sentence. In this case we have two atomic sentences, I and H, so we have four possible valuations, each of which is a line in the truth table:

I	H	$(\neg I \,\&\, H) \to H$
T	T	
T	F	
F	T	
F	F	

Our job is to fill out the truth values of $(\neg I \,\&\, H) \to H$.

Here the formation tree will help us know what to do (see §5.1):



The truth rule for $\neg$ tells us how the truth value of $\neg I$ depends on the truth of I. Then the rule for & tells us how the truth value of $\neg I \,\&\, H$ depends on the truths of $\neg I$ and H; and finally, the rule for $\to$ tells us how the truth value of $(\neg I \,\&\, H) \to H$ depends on those of $\neg I \,\&\, H$ and H.

So to work out the truth values of $(\neg I \,\&\, H) \to H$ we first need to work out the truth values of $\neg I$ and $\neg I \,\&\, H$.

So, we expand our truth table with columns for each of these.

I	H	$\neg I$	$(\neg I \,\&\, H)$	$(\neg I \,\&\, H) \to H$
T	T			
T	F			
F	T			
F	F			

The first step is $\neg I$. To be able to do this, we will use the truth rule we specified for $\neg$:

If X is		then $\neg X$ is
T	$\rightsquigarrow$	F
F	$\rightsquigarrow$	T

Now, we can fill out:

I	H	$\neg I$	$(\neg I \,\&\, H)$	$(\neg I \,\&\, H) \to H$
T	T	a=F		
T	F	b=F		
F	T	c=T		
F	F	d=T		
★				

We worked these out by:

- For 'a': Look at the column for I, and see it's T, so by our rule (T-line), we fill out a=F.
- For 'b': Look at the column for I, and see it's T, so by our rule (T-line), we fill out b=F.
- For 'c': Look at the column for I, and see it's F, so by our rule (F-line), we fill out c=T.
- For 'd': Look at the column for I, and see it's F, so by our rule (F-line), we fill out d=T.

The next step is to consider $\neg I \,\&\, H$. For this we will use the truth rule for &:

If X is	and Y is		then $X \,\&\, Y$ is
T	T	$\rightsquigarrow$	T
T	F	$\rightsquigarrow$	F
F	T	$\rightsquigarrow$	F
F	F	$\rightsquigarrow$	F

Now, we can fill out:

I	H	$\neg I$	$(\neg I \,\&\, H)$	$(\neg I \,\&\, H) \to H$
T	T	F	a=F	
T	F	F	b=F	
F	T	T	c=T	
F	F	T	d=F	
	★	★		

We worked these out by:

▹ For ‘a’: Look at the column for $\neg I$ and the column for H, we have F and T; so by our rule (FT-line), we fill out a=F.

▹ For ‘b’: Look at the column for $\neg I$ and the column for H, we have F and F; so by our rule (FF-line), we fill out a=F.

▹ For ‘c’: Look at the column for $\neg I$ and the column for H, we have T and T; so by our rule (TT-line), we fill out a=T.

▹ For ‘d’: Look at the column for $\neg I$ and the column for H, we have T and F; so by our rule (TF-line), we fill out a=F.

Now, finally, we need to look at $(\neg I \,\&\, H) \rightarrow H$, and will use the truth rule for $\rightarrow$:

If X is	and Y is		then $X \rightarrow Y$ is
T	T	$\rightsquigarrow$	T
T	F	$\rightsquigarrow$	F
F	T	$\rightsquigarrow$	T
F	F	$\rightsquigarrow$	T

Now, we can fill out:

I	H	$\neg I$	$(\neg I \,\&\, H)$	$(\neg I \,\&\, H) \rightarrow H$
T	T	F	F	a=T
T	F	F	F	b=T
F	T	T	T	c=T
F	F	T	F	d=T
	★consequent		★antecedent	

We worked these out by:

▹ For ‘a’: In the column for $(\neg I \,\&\, H)$ (which is our antecedent) we have F; and in the column for H (our consequent) we have T. For the conditional, it’s very important to bear in mind which order they come in. We are looking at antecedent then consequent, so the rule line we are looking at is FT (the antecedent is False and the consequent is True rather than visa versa). And we get a=T.

▹ For ‘b’: In the column for $(\neg I \,\&\, H)$ we have F; and in the column for H we have F. So by our rule for $\rightarrow$ (FF-line) we have b=T.

▹ For ‘c’: In the column for $(\neg I \,\&\, H)$ we have T; and in the column for H we have T. So by our rule for $\rightarrow$ (TT-line) we have b=T.

- ▹ For 'd': In the column for $(\neg I \,\&\, H)$ we have F; and in the column for H we have F. So by our rule for $\rightarrow$ (FF-line) we have d=T.

When we do these in practice, our sentences can have many subsentences, consider, e.g.

$$\neg(B \,\&\, (\neg B \leftrightarrow \neg A)))$$

A tool for fitting our truth tables on a page is not to put all the subsentences out as separate entire columns to calculate, but to simply list the values underneath the main connective of the sentence. This is simply to keep the truth table more concise.

So instead of writing

A	B	$\neg B$	$\neg A$	$\neg B \leftrightarrow \neg A$	$(B \,\&\, (\neg B \leftrightarrow \neg A)))$	$\neg(B \,\&\, (\neg B \leftrightarrow \neg A)))$
T	T	F	F	T	T	F
T	F	T	F	F	F	T
F	T	F	T	F	F	T
F	F	T	T	T	F	T

I can write:

A	B	$\neg$	$(B$	$\&$	$(\neg B$	$\leftrightarrow$	$\neg A)))$
T	T	**F**		T	F	T	F
T	F	**T**		F	T	F	F
F	T	**T**		F	F	F	T
F	F	**T**		F	T	T	T

and highlight the column under the main connective, which provides our final answer. You should feel free to use whichever method you find easier.

There are tricks that mean you can miss out some of the gaps by using strategies such as: knowing that X is false is already enough to see that $X \,\&\, Y$ is false, so we don't need to continue to work out the truth value of Y. These "shortcuts" are discussed in §14, but we won't read it in this course.

11.2 The possible valuations

The size of the complete truth table depends on the number of different atomic sentences in the table. A sentence that contains only one atomic sentence requires only two rows, as in the characteristic truth table for negation. This is true even if the same letter is repeated many times, as in the sentence '$[(C \leftrightarrow C) \rightarrow C] \,\&\, \neg(C \rightarrow C)$'. The complete truth table requires only two lines because there are only two possibilities: 'C' can be true or it can be false. The truth table for this sentence looks like this:

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

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

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

A sentence that contains three atomic sentences requires eight lines:

M	N	P	$M \,\&\, (N \vee P)$
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 '$M \,\&\, (N \vee P)$' can be true or false, depending on the truth values of 'M', 'N', and 'P'.

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

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

11.3 More about brackets

Consider these two sentences:

$$((A \,\&\, B) \,\&\, C)$$
$$(A \,\&\, (B \,\&\, C))$$

These are truth functionally equivalent. Consequently, it will never make any difference from the perspective of truth value – which is all that TFL cares about (see §10) – which of the two sentences we assert (or deny). Even though the order of the brackets does not matter as to their truth, we should not just drop them. The expression

$$A \,\&\, B \,\&\, C$$

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

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

But we should not simply write:

$$A \vee B \vee C$$

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

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

So if we were to write:

$$A \rightarrow B \rightarrow C$$

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

$$((A \lor B) \,\&\, C)$$
$$(A \lor (B \,\&\, C))$$

So if we were to write:

$$A \lor B \,\&\, C$$

it would be dangerously ambiguous. *Never write this.* The moral is: never drop brackets.

CHAPTER 12

Validity in TFL

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

12.1 Validity

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

Now, since we are going to be talking about arguments a lot, we will introduce some abbreviation. An argument had the form:

- X_1
- X_2
- $\ldots$
- X_n
- Therefore: Y

We can write this more concisely as:

$$X_1, X_2, \ldots, X_n \therefore Y$$

The $\therefore$ symbol can be read out loud as ‘therefore’. (When typing any answers out in word, you can replace it with ‘Therefore’.)

This is an argument, which may be valid or not.

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

> The TFL argument ‘$X_1, X_2, \ldots, X_n \therefore Y$’ is VALID iff there are no valuations where the premises are true and the conclusion false.

Validity is a property of an *argument*. If we want to talk about the conclusion being a consequence of the premises, or *following from*, we use the term ‘*entails*’.

> The TFL sentences $X_1, X_2, \ldots, X_n$ ENTAIL the sentence Y iff there are no valuations where all of $X_1, X_2, \ldots, X_n$ true and Y false.

An argument is valid if and only if its premises entail its conclusion.

It is easy to check for this with a truth table. Consider the argument:

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

We need to check whether there is any valuation which makes both $\neg L \rightarrow (J \vee L)$ and $\neg L$ true whilst making J false. So we use a truth table:

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

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

In short, we have a way to test for the validity of English arguments. First, we symbolize them in TFL, as having premises $X_1, X_2, \ldots, X_n$, and conclusion Y. Then we check whether they are valid using truth tables.

‘Entails’ versus ‘$\rightarrow$’

We now want to compare and contrast ‘entails’ and ‘$\rightarrow$’.

Observe: X entails Y iff there is no valuation of the atomic sentences that makes X true and Y false.

Observe: $X \rightarrow Y$ is a tautology iff there is no valuation of the atomic sentences that makes $X \rightarrow Y$ false. Since a conditional is true except when its antecedent is true and its consequent false, $X \rightarrow Y$ is a tautology iff there is no valuation that makes X true and Y false.

Combining these two observations, we see that $X \rightarrow Y$ is a tautology iff X entails Y. But there is a really, really important difference between entailment and ‘$\rightarrow$’: ‘$\rightarrow$’ is a sentential connective of TFL. ‘Entails’ is a word in English. When ‘$\rightarrow$’ is flanked with two TFL sentences, the result is a longer TFL sentence. By contrast, when we use ‘entails’, we are expressing that there is a relationship between the surrounding TFL sentences.

12.2 Formal validity and English arguments

We can now use the tool of TFL to investigate whether an argument of English is formally valid.

Consider an argument like:

> If Jones signed the contract under duress, then the contract is void. But since it was not signed under duress, the contract is not void.[1]

We are interested in determining if this is valid or not. Here’s a general strategy to help us:

[1] This sort of example is discussed in https://lawpublications.barry.edu/cgi/viewcontent.cgi?article=1026&context=barrylrev.

1. Find the structure of the argument.
   Identify the premises and conclusion.

2. Symbolise the argument in TFL.

3. Check if the TFL argument is valid.

   ▹ Using truth tables to look for a valuation providing a counter example. If there is no such valuation, then it is valid.

   ▹ Or, use natural deduction to show that it is valid.

Following this procedure, then, we need to first find the structure of the argument. It has two premises and a conclusion:

- If Jones signed the contract under duress, then the contract is void.
- Jones did not sign the contract under duress.
- Therefore: The contract is not void.

We are interested in whether it is valid. I.e. is there a situation where the premises are true and the conclusion is false. Perhaps in this case you can work it out, but if not TFL provides us a tool for thinking through all the possibilities. We therefore move to stage 2 and symbolise the sentences of the argument in TFL as well as we can.

- $D \rightarrow V$
- $\neg D$
- Therefore: $\neg V$

And moving to stage 3, we check whether the TFL argument is valid. We do this with truth tables. Later in the course we'll see Natural Deduction which will provide an alternative method for checking validity.

So we need to fill in truth table with all the sentences involved in the argument. In this case each sentence has a simple structure, if we draw out the formation trees they just have one step, e.g.

$$\begin{tikzcd}[arrows=dash] & D \rightarrow V \arrow[dl] \arrow[dr] & \\ D & & V \end{tikzcd}$$

So our truth table doesn't need any additional "calculation" lines. We can directly fill it in, following our truth rules

		Premise	Premise	Concln	
D	V	$D \to V$	$\neg D$	$\neg V$	
T	T	T	F	F	
T	F	F	F	T	
F	T	T	T	F	⇜ counterexample
F	F	T	T	T	

And we can see that this argument is invalid. The valuation providing the counter example is line 3: $\begin{array}{cc} D & V \\ \hline \mathrm{F} & \mathrm{T} \end{array}$

This allows us to read of a possibility where the premises are true but the conclusion false: It might be that the contract is void without having been made under duress. E.g. mental illness, misrepresentation,....

This has then allowed us to see that the original English argument is invalid. In fact, the argument followed an invalid pattern that is commonly found. It is a common fallacy called "denying the consequent".

This argument is invalid. So we can accept the premises and still reject the conclusion. If we have a valid argument, then the conclusion can only be rejected if the premises are rejected. In a valid argument our discussion needs to be focused on the premises.

In fact in this case we can give a fixed-up, related, valid argument

- $V \to D$
- $\neg D$
- Therefore: $\neg V$

This uses the $\to$ the other way around and is valid. It is an an argument form called Modus Tollens. However, presenting this argument and showing it's valid isn't enough to convince me of the conclusion that the contract is void: the argument uses a false premise: $V \to D$. Or at least, this is a premise that I won't accept without further justification: look, if there was misrepresentation in the trial then $V \to D$ is false. Give me a way of ruling out misrepresentation if you want to get to the conclusion that the contract is not void.

Another example: disjunctive reasoning

> The murder weapon was the dagger or the rope. So Mrs. Peacock is guilty. After all, if the murder weapon was

the dagger then she's guilty. And if it was the rope, she's guilty.

First we find the argument structure:

- The murder weapon was the dagger or the rope.
- If the murder weapon was the dagger then Mrs. Peacock is guilty.
- If the murder weapon was the rope then Mrs. Peacock is guilty.
- Therefore: Mrs. Peacock is guilty.

We then symbolise each constituent sentence into TFL:

- $D \vee R$
- $D \rightarrow G$
- $R \rightarrow G$
- Therefore: G

Now, we check if the TFL argument is valid using truth tables:

			Premise	Premise	Premise	Concln
D	G	R	$D \vee G$	$D \rightarrow G$	$R \rightarrow G$	G
T	T	T	T	T	T	T
T	T	F	T	T	T	T
T	F	T	T	F	F	F
T	F	F	T	F	T	F
F	T	T	T	T	T	T
F	T	F	T	T	T	T
F	F	T	F	T	F	F
F	F	F	F	T	T	F

And we see that this argument is valid: there is no valuation (=line) where the premises are all true and the conclusion false.

12.3 Other kinds of validity

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

This allows us to investigate the formal validity of an argument. More precisely, whether its TFL-form is a valid form.

If the TFL-form of an argument is valid, then the argument is also conceptually valid, nomologically valid, and valid in any other (plausible) sense. We cannot conceive of possibilities that do not follow the laws of logic.

If the TFL symbolisation of an argument is valid (in TFL), then the original argument is formally valid, and valid in all the other senses.

However, some arguments may have their TFL forms being invalid but are nonetheless conceptually valid. For example:

- That is a triangle.
- Therefore: That has three sides.

This argument is conceptually valid. But it would be symbolised in TFL simply as:

- A
- Therefore: B

and looking at the truth tables (which are very easy to do!) we see:

		Premise	Concln	
A	B	A	B	
T	T	T	T	
T	F	T	F	⇜ counterexample valuation, but not conceptually possible
F	T	F	T	
F	F	F	F	

The valuation in line 2 gives us a counterexample: it makes the premise, A, true, and the conclusion, B, false. This valuation, however, violates the conceptual connections between words in English. Triangles simply have to have three sides. The "case" being described by this valuation makes it true that it is a triangle but false that it has three sides. That is not a conceptual possibility. The argument is conceptually valid, but its TFL symbolisation is invalid.

If the TFL symbolisation is valid, the argument must be conceptually valid too. But some conceptually valid arguments have invalid TFL symbolisations.

Consider:

- If Jones signed the contract under duress, then the contract is void.
- Jones did not sign the contract under duress.
- Therefore: The contract is not void.

which we symbolised in TFL as:

$$D \to V, \ \neg D \therefore \neg V$$

which is invalid. The valuation providing the counterexample is: $\begin{array}{cc} D & V \\ \hline \text{F} & \text{T} \end{array}$ To see if the argument is *conceptually* valid, one should then investigate whether the valuations providing counterexamples to the formal validity are conceivable possibilities. Here, we're probably interested in something like legal validity, which governs conditions under which a contract is void. We now know we should investigate whether the possibility of it being void but not signed under duress is compatible with the legal laws. This will depend on what the legal laws are. Actually, a contract can be void without being signed under duress, for example due to misrepresentation or mental incapacity. So the argument is in fact legally invalid as well.

There is one loose end to tie up. In fact, TFL is in't all there is to formal validity. If the TFL symbolisation of an argument is valid, then the original argument is formally valid. But consider the following argument: Consider

- Socrates is a man.
- All men are mortal.
- Therefore: Socrates is mortal.

When we symbolise this in TFL, each of the sentences is simply an atomic sentence, so we get:

$$A, B \ \therefore \ C$$

which is invalid. The problem isn't the symbolisation we gave. It is the best we could do *in TFL*. The problem is that TFL is limited in the "form" it can see, after all. We will improve on this when we move to First Order Logic in §24. This argument is formally valid in virtue of its FOL form rather than TFL form.

CHAPTER 13

Other logical notions

13.1 Tautologies and contradictions

We have similar substitutes for the notions we introduced in §2. There we explained *necessary truth* and *necessary falsity*. Both notions have surrogates in TFL. We will start with a surrogate for necessary truth.

> X is a TAUTOLOGY iff it is true on every valuation.

We can determine whether a sentence is a tautology just by using truth tables. If the sentence is true on every line of a complete truth table, then it is true on every valuation, so it is a tautology. In the example of §11, '$(H \,\&\, I) \rightarrow H$' is a tautology.

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

A	$A \lor \neg A$
T	T **T** F T
F	F **T** T F

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

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

We have a similar surrogate for necessary falsity:

> X is a CONTRADICTION iff it is false on every valuation.

We can determine whether a sentence is a contradiction just by using truth tables. If the sentence is false on every line of a complete truth table, then it is false on every valuation, so it is a contradiction.

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

13.2 Logical equivalence

Here is a similar useful notion:

> X and Y are LOGICALLY EQUIVALENT iff, for every valuation, their truth values agree, i.e. if there is no valuation in which they have opposite truth values.

We have already made use of this notion, in effect, in §11.3; the point was that '$(A \,\&\, B) \,\&\, C$' and '$A \,\&\, (B \,\&\, C)$' are logically equivalent. Again, it is easy to test for logical equivalence using truth tables. Consider the sentences '$\neg(P \vee Q)$' and '$\neg P \,\&\, \neg Q$'. Are they logically equivalent? To find out, we construct a truth table.

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

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

13.3 Consistency

In §2, we said that sentences are jointly possible iff it is possible for all of them to be true at once. We can offer a surrogate for this notion too:

> $X_1, X_2, \ldots, X_n$ are CONSISTENT iff there is some valuation which makes them all true.

Derivatively, sentences are jointly logically inconsistent if there is no valuation that makes them all true. Again, it is easy to test for joint logical consistency using truth tables.

13.4 These notions and the English variants

Just as in §12.2 we said we can use the logical notion of validity in TFL to help with various notions of validity of English, the same can be said for these other logical notions.

For example, if you show that the TFL symbolisation of an English sentence is a tautology, you can conclude that it is a necessary truth. However, there are different kinds of 'necessary truths'. Some sentences like 'A triangle has three sides' are necessary truths, but are not tautologies. But the tool of TFL can help.

13.5 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	$\neg$	$(P$	$\&$	$\neg Q)$	$\&$	$\neg P$
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	$((P$	$\rightarrow$	$Q)$	$\rightarrow$	$P)$	$\rightarrow$	P
T	T						**T**	
T	F						**T**	
F	T		T		F		**T**	
F	F		T		F		**T**	

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

13.6 Testing for validity and entailment

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

CHAPTER 14

Partial truth tables

Not taught in course. This knowledge will not be needed for the exam. (Although you can use these techniques in the exam if you wish.)

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

Tautology. In order to show that a sentence is a tautology, we need to show that it is true on every valuation. That is to say, we need to know that it comes out true on every line of the truth table. So we need a complete truth table.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

This table summarises what is required:

	Yes	**No**
tautology?	complete truth table	one-line partial truth table
contradiction?	complete truth table	one-line partial truth table
equivalent?	complete truth table	one-line partial truth table
consistent?	one-line partial truth table	complete truth table
valid?	complete truth table	one-line partial truth table
entailment?	complete truth table	one-line partial truth table

CHAPTER 15

TFL vs English connectives

Not covered in this course. This is just a taster for students who are interested.

Consider the sentence:

1. Jan is neither bald nor not-bald.

To symbolize this sentence in TFL, we would offer something like '$\neg J \,\&\, \neg\neg J$'. This a contradiction (check this with a truth-table), but sentence 1 does not itself seem like a contradiction; for we might have happily go on to add 'Jan is on the borderline of baldness'!

Third, consider the following sentence:

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

Symbolizing this in TFL, we would offer something like '$\neg(G \to M)$'. Now, '$\neg(G \to M)$' entails 'G' (again, check this with a truth table). So if we symbolize sentence 2 in TFL, it seems to entail that God exists. But that's strange: surely even an atheist can accept sentence 2, without contradicting herself!

One lesson of this is that the symbolization of 2 as '$\neg(G \rightarrow M)$' shows that 2 does not express what we intend. Perhaps we should rephrase it as

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

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

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

PART IV

Natural deduction for TFL

CHAPTER 16

The very idea of natural deduction

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

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

However, providing a truth table is not how one will usually work with arguments. When you actually use arguments, for example in philosophical essays, you typically instead will break down an argument into smaller steps. If you are trying to show to your reader that a particular argument is valid, the truth table method says to check all possibilities. We will instead provide an alternative way of showing arguments are valid which matches natural ways of reasoning.

The idea of natural deduction is how you might go about trying to convince an interlocutor that your argument is valid. You would generally do this by breaking your argument into smaller steps that are obviously valid and piecing these together.

Suppose you're trying to convince someone that

$$A \to (B \,\&\, C), A \therefore B$$

is valid, and they don't see it. You can help them by breaking it up into two steps: first see that $B \,\&\, C$ follows and then note that B follows from $B \,\&\, C$.

In more detail: Grant me that $A \to (B \,\&\, C)$ and A are true. Then what else do we know to be true? Here's something: $B \,\&\, C$. Why? Because in general Modus Ponens is an excellent argument pattern: any argument of the form X, $X \to Y \quad \therefore \quad Y$ is valid: there are no valuations where X and $X \to Y$ are true but Y is false. So if our premises $A \to (B \,\&\, C)$ and A are true, then $B \,\&\, C$ must also be true: that's just Modus Ponens. So now from our supposition of $A \to (B \,\&\, C)$ and A, we now also know $B \,\&\, C$. What else follows from these three statements? Here's something: B. Why? Well, $B \,\&\, C$ is true; so certainly B must be true. So we know that from $A \to (B \,\&\, C)$ and A we can conclude B by walking someone through these two steps. This will be enough to show that $A \to (B \,\&\, C)$, $A \therefore B$ is valid.

To keep track of what assumptions have been made and steps of the argument we will give precise forms that this argument should be written:

1	$A \to (B \,\&\, C)$	
2	A	
3	$B \,\&\, C$	From 1, 2
4	B	From 3

Our premises are written above the horizontal line. They have to be granted without justification. Then each new line follows from the previous lines. The vertical line is there to highlight that everything coming below is within the context of the premises that have been assumed; that we are looking for consequences of the premises.

We can also use this presentation to be clear about arguments that we make in English:

1	If Alice came to the party, then Beth and Cath came	
2	Alice came to the party	
3	Beth and Cath came to the party	From 1, 2
4	Beth came to the party	From 3

You might think of it as a bag you're collecting things to be accepted in. You have to grant the premises, they go in the bag for free, then we give certain rules that allow us to add additional statements which must be true so long as the other things already in the bag are true.

Suppose I provide you with the following argument:

1	$P \to (\neg Q \to \neg R)$	
2	$P \to \neg Q$	
3	$P \,\&\, S$	
4	$\neg R$	From 1, 2, 3

I concluded line 3 as a logical consequence of lines 1, 2 and 3. It does follow, i.e, the argument is valid; but this is not very helpful to someone who doesn't yet see that it's valid.

Instead, we will be describing various rules which we propose that have to be accepted as valid reasoning steps, and all more complicated steps should be broken up into simpler ones. So we should break this argument up into the steps:

1	$P \to (\neg Q \to \neg R)$	
2	$P \to \neg Q$	
3	$P \,\&\, S$	
4	P	From 3
5	$\neg Q \to R$	From 1, 4
6	$\neg Q$	From 2, 4
7	R	From 5, 6

The other thing we should do is to give a name for the steps that we use. Here we've just said 'From 1,4', but someone might ask: *how* does it follow from lines 1 and 4. We will give names for the various simple steps of reasoning we use and say: “well, it follows from lines 1 and 4 by the rule “→E”.”

1	$P \to (\neg Q \to \neg R)$	
2	$P \to \neg Q$	
3	$P \,\&\, S$	
4	P	&E 3
5	$\neg Q \to R$	→E 1, 4
6	$\neg Q$	→E 2, 4
7	R	→E 5, 6

We will provide various rules, and describe why they are acceptable. We should then break any other valid arguments should be broken up into these steps of reasoning.

16.1 More reasons for natural deduction

Using truth tables to show validity does not necessarily give us much *insight*. Consider two arguments in TFL:

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

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

One aim of a *natural deduction system* is to show that particular arguments are valid, in a way that allows us to understand the reasoning that the arguments might involve.

This is a very different way of thinking about arguments.

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

The move to natural deduction might be motivated by more than the search for insight. It might also be motivated by *necessity*. Once our arguments involve 5 atomic sentences, a truth table test for validity will require 32 lines of truth table. That's quite a lot to check. But sometimes we might want to check such arguments.

- Alice, or Betty, or Carys, or Dan, or Ella stole the teacher's pen.
- It wasn't Alice.
- It wasn't Betty,
- It wasn't Carys,
- It wasn't Dan

∴ It was Ella.

$$A \lor (B \lor (C \lor (D \lor E))), \neg A, \neg B, \neg C, \neg D \;\therefore\; E$$

And that will increase exponentially as more atomic sentences get added. Once an argument involves 20 atomic sentences,

- Alice, or Betty, or Carys, . . . , or Uli or Volker stole the teacher's pen.
- It wasn't Alice.
- It wasn't Betty,
- $\vdots$
- It wasn't Uli,
- Therefore: It was Volker.

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

When we get to first-order logic (FOL) (beginning in chapter 24) the problem gets dramatically worse. There is nothing like the truth table test for FOL. To assess whether or not an argument is valid, we have to reason about *all* interpretations, but, as we will see, there are infinitely many possible interpretations. We cannot even in principle set a machine to grind through infinitely many possible interpretations and report back when it is finished: it will *never* finish. We either need to come up with some more efficient way of reasoning about all interpretations, or we need to look for something different. We will be looking for something different; and we will develop natural deduction.[1]

The modern development of natural deduction dates from simultaneous and unrelated papers by Gerhard Gentzen and Stanisław Jaśkowski (both in 1934). However, the natural deduction system that we will consider is based largely around work by Frederic Fitch (first published in 1952).

Natural deduction selects a few basic rules of inference and natural forms of reasoning and encodes these into a proof system. We will now see natural deduction for TFL. This system will form the basis also for natural deduction for FOL, which will also add rules for the quantifiers.

[1] There are, in fact, systems that codify ways to reason about all possible interpretations which can be used for FOL in a similar way to the way we use truth tables for TFL. They were developed in the 1950s by Evert Beth and Jaakko Hintikka, but we will not follow this path.

CHAPTER 17

The First Basic Rules for TFL

We will now describe the various rules one can use. All other valid arguments should be broken up into steps using these rules. We will give a particular list of rules, other systems will choose other particular rules.

The rules we give will mostly be attached to particular connectives. This is something that helps with the way to go about finding proofs and it has some technical benefits.

17.1 Reiteration

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

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

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

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

Here is a general expression of the rule:

$$\begin{array}{r|ll} & \vdots & \\ m & X & \\ & \vdots & \\ & X & \quad \text{R } m \end{array}$$

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

Two things need emphasising. First 'X' is not a sentence of TFL. Rather, it a symbol in the metalanguage, which we use when we want to talk about any sentence of TFL (see §8). Second, and similarly, 'm' is not a numeral that will appear on a proof. Rather, it is a symbol in the metalanguage, which we use when we want to talk about any line number of a proof. In an actual proof, the lines are numbered '1', '2', '3', and so forth. But when we define the rule, we use variables to underscore the point that the rule may be applied at any point.

Why might this be useful? For example, we can now show $A \therefore A$ is valid using the proof:

$$\begin{array}{r|ll} 1 & A & \\ \cline{2-2} 2 & A & \quad \text{R } 2 \end{array}$$

The rule really becomes useful, though, once we are dealing with subproofs, which we will see in the next chapter.

17.2 Modus Ponens

Consider the following argument:

> If Jane is smart then she is fast. Jane is smart. $\therefore$ Jane is fast.

This argument is certainly valid. In fact any argument of the form

$$X \to Y, X \;\therefore\; Y$$

is valid. We introduce a rule of natural deduction that encodes this idea. This is called *Modus Ponens.*

We introduce a rule of Natural Deduction which allows us to make this reasoning step. We will call it the "Conditional Elimination" rule ($\to$E). This choice of name is because we start with something including the connective $\to$ and we derive something without the connective, that is we have *eliminated* the $\to$ connective. For each connective we will have introduction and elimination rules, however we will wait until the next chapter to see Conditional Introduction.

In a simple use of this rule, we might just use it to derive from the premises $S \to F$ and S the conclusion F:

1	$S \to F$	
2	S	
3	F	$\to$E 1, 2

This would then be a natural deduction proof that $S \to F, S \;\therefore\; F$ is valid.

Each line, except for the premsies which are taken as assumptions, has to be labelled with the rule it used. So here, we write "$\to$E 1,2" to say that we obtained line 3 by use of this rule $\to$ Elimination applied to lines 1 and 2.

We can also apply the rule when our $X \to Y$ and X are not themselves premises but have themselves been derived in the course of the proof.

$$
\begin{array}{r|ll}
1 & \text{Premise 1} & \\
2 & \text{Premise 2} & \\
\hline
 & \vdots & \\
8 & S \to F & \text{some rule} \\
 & \vdots & \\
15 & S & \text{another rule} \\
 & \vdots & \\
23 & F & \to\text{E } 8, 15
\end{array}
$$

It also can be that they appear in a different order, or that one appears in the premises, for example:

$$
\begin{array}{r|ll}
1 & S & \\
\hline
 & \vdots & \\
8 & S \to F & \\
 & \vdots & \\
23 & F & \to\text{E } 8, 1
\end{array}
$$

We write our general rule as:

$$
\begin{array}{r|ll}
 & \vdots & \\
m & X \to Y & \\
 & \vdots & \\
n & X & \\
 & \vdots & \\
 & Y & \to\text{E } m, n
\end{array}
$$

We can apply it to any X and Y. For example,

1	$(A \vee B) \rightarrow \neg F$	
2	$(A \vee B)$	
3	$\neg F$	$\rightarrow$E 1, 2

In this, X is $(A \vee B)$, and Y is $\neg F$.

We would typically now move to introducing Conditional Introduction. However, we will first do all the other rules of the system, because Conditional Introduction involves additional complexity.

17.3 Conjunction Introduction

Suppose we want to show that Alice and Beth both came to the party. One obvious way to do this would be as follows: first we show that Alice came to the party; then Beth came to the party; then we put these two demonstrations together, to obtain the conjunction.

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

A: Alice came to the party
B: Beth came to the party

Perhaps we are working through a proof, and we have obtained 'A' on line 8 and 'B' on line 15. Then on any subsequent line we can obtain '$A \& B$'. For example our proof might contain the following lines:

8	A	
15	B	
23	$A \& B$	&I 8, 15

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

	$\vdots$	
m	X	
	$\vdots$	
n	Y	
	$\vdots$	
	$X \,\&\, Y$	&I m, n

Two things need emphasising.

First 'X' and 'Y' are metavariables. They are not particular sentences of TFL but are there to play the role of any particular sentence (see §8).

Similarly, 'm' is not a numeral that will appear on a proof. Rather, it is a symbol in the metalanguage, which we use when we want to talk about any line number of a proof. In an actual proof, the lines are numbered '1', '2', '3', and so forth. But when we define the rule, we use variables to underscore the point that the rule may be applied at any point.

To be clear, the statement of the rule is *schematic*. It is not itself a proof. 'X' and 'Y' are not sentences of TFL. Rather, they are symbols in the metalanguage, which we use when we want to talk about any sentence of TFL (see §8). Similarly, 'm' and 'n' are not a numerals that will appear on any actual proof. Rather, they are symbols in the metalanguage, which we use when we want to talk about any line number of any proof. In an actual proof, the lines are numbered '1', '2', '3', and so forth, but when we define the rule, we use variables to emphasize that the rule may be applied at any point. The rule requires only that we have both conjuncts available to us somewhere in the proof. They can be separated from one another, and they can appear in any order.

The rule is called 'conjunction *introduction*' because it introduces the symbol '&' into our proof where it may have been absent.

17.4 Conjunction Elimination

Correspondingly, we have a rule that *eliminates* that symbol. Suppose you have shown that Alice and Beth both came to the party. You are entitled to conclude that Alice came to the party. Equally, you are entitled to conclude that Beth came to the party. Putting this together, we obtain our conjunction elimination rule(s):

$$\begin{array}{r|ll} & \vdots & \\ m & X \,\&\, Y & \\ & \vdots & \\ & X & \&\text{E } m \end{array}$$

and equally:

$$\begin{array}{r|ll} & \vdots & \\ m & X \,\&\, Y & \\ & \vdots & \\ & Y & \&\text{E } m \end{array}$$

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

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

- $[(A \lor B) \rightarrow (C \lor D)] \,\&\, \neg(E \lor F)$
- Therefore: $\neg(E \lor F) \,\&\, [(A \lor B) \rightarrow (C \lor D)]$

The main logical operator 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 (repeated applications of) our rules of inference. So the beginning of the proof looks like this:

$$\begin{array}{r|l} 1 & \underline{[(A \lor B) \rightarrow (C \lor D)] \,\&\, \neg(E \lor F)} \end{array}$$

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

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

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

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

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

It is worth giving another example. Way back in §11.3, we noted that this argument is valid:

$$A \,\&\, (B \,\&\, C) \therefore (A \,\&\, B) \,\&\, C$$

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

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

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

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

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

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

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

Let me offer one final illustration. When using the &I rule, there is no requirement that it is applied to two different sentences. So we can formally prove 'A' from 'A' as follows:

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

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

17.5 Biconditional

For the biconditional we encode the idea that $X \leftrightarrow Y$ is equivalent to $(X \rightarrow Y) \,\&\, (Y \rightarrow X)$.

The rules we give, then, are very closely related to the rules for conjunction: Two elimination rules

$$
\begin{array}{r|ll}
 & \vdots & \\
m & X \leftrightarrow Y & \\
 & \vdots & \\
 & X \rightarrow Y & \leftrightarrow\text{E } m
\end{array}
$$

and

$$
\begin{array}{r|ll}
 & \vdots & \\
m & X \leftrightarrow Y & \\
 & \vdots & \\
 & Y \rightarrow X & \leftrightarrow\text{E } m
\end{array}
$$

and an introduction rule:

$$
\begin{array}{r|ll}
 & \vdots & \\
m & X \rightarrow Y & \\
 & \vdots & \\
n & Y \rightarrow X & \\
 & \vdots & \\
 & X \leftrightarrow Y & \leftrightarrow\text{I } m, n
\end{array}
$$

17.6 Disjunction Introduction

Suppose Alice came to the party. Then Alice or Beth came to the party. After all, to say that Alice or Beth came to the party is to say something weaker than to say that Alice came to the party.

Let me emphasize this point. Suppose Alice came to the party. It follows that Alice came to the party *or* I am the Queen of England. Equally, it follows that Alice or the Queen came to the party. Equally, it follows that Alice came to the party *or* that God is dead. Many of these are strange inferences to draw, but there is nothing *logically* wrong with them (even if they maybe violate all sorts of implicit conversational norms).

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

$$\begin{array}{r|ll} & \vdots & \\ m & X & \\ & \vdots & \\ & X \vee Y & \vee\mathrm{I}\ m \end{array}$$

and

$$\begin{array}{r|ll} & \vdots & \\ m & X & \\ & \vdots & \\ & Y \vee X & \vee\mathrm{I}\ m \end{array}$$

Notice that Y can be *any* sentence whatsoever, so the following is a perfectly acceptable proof:

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

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

17.7 Law of Excluded Middle

We will actually add another rule for how to introduce a disjunction. There are special kinds of disjunctions that don't need further justification: sentences of the form $X \vee \neg X$. In §13.1 we saw that $A \vee \neg A$ is a tautology: it is true on all valuations. More generally, any sentence of the form $X \vee \neg X$ is a tautology. The rule *Law of Excluded Middle* encodes this fact: it simply says that you are always allowed to write $X \vee \neg X$:

$$
\begin{array}{l|ll}
 & \vdots & \\
 & X \vee \neg X & \text{LEM}
\end{array}
$$

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

To see the rule in action, consider:

$$A \to B, \neg A \to B \ \therefore B$$

We can give the following proof to show this is valid:

$$
\begin{array}{r|ll}
1 & A \to B & \\
2 & \neg A \to B & \\
\hline
3 & A \vee \neg A & \text{LEM} \\
4 & B & \vee\text{E } 1, 2, 3
\end{array}
$$

The law of Excluded Middle is often used in combination with Disjunction Elimination.

However, we will not yet introduce Disjunction Elimination. It will be introduced in §18.2

17.8 Contradiction

Instead of tackling negation directly, we will first think about *contradiction.*

Contradiction Introduction

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

whenever we find both a sentence and its negation appearing in our proof:

$$
\begin{array}{r|ll}
 & \vdots & \\
m & X & \\
 & \vdots & \\
n & \neg X & \\
 & \vdots & \\
 & \bot & \bot\mathrm{I}\ m, n
\end{array}
$$

It does not matter what order the sentence and its negation appear in, and they do not need to appear on adjacent lines.

Contradiction Elimination

Our elimination rule for '⊥' is known as *ex falso quod libet*, or *explosion*. This means 'anything follows from a contradiction', and the idea is precisely that: if we obtained a contradiction, symbolized by '⊥', then we can infer whatever we like. How can this be motivated, as a rule of argumentation? Well, consider the English rhetorical device '…and if *that's* true, I'll eat my hat'. Since contradictions simply cannot be true, if one *is* true then not only will I eat my hat, I'll have it too.[1] Here is the formal rule:

$$
\begin{array}{r|ll}
 & \vdots & \\
m & \bot & \\
 & X & \bot\mathrm{E}\ m
\end{array}
$$

Note that X can be *any* sentence whatsoever.

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

[1] Thanks to Adam Caulton for this.

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

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

17.9 Strategies

We have a few more rules to introduce, but at this point we pause to mention a strategies to come up with proofs yourself:

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

For example: If your conclusion is a conditional $X \& Y$, plan to use the &I rule. This requires finding both X and Y.

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 often 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.

Sometimes, though, this won't yet be possible. For that, we need to finish off our list of the basic rules of natural deduction.

CHAPTER 18

More Basic Rules for TFL

To introduce a conditional we need to introduce another kind of thing: a sub-proof.

18.1 Conditional Introduction

The following argument is valid:

> Alice came to the party. Therefore if Beth came to the party, then both Alice and Beth came.

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

> Assume that Alice came to the party. Now, *additionally* assume that Beth came to the party. Then by conjunction introduction—which we just discussed—both Alice and Beth came. Of course, that's conditional on the assumption that Beth came to the party. But this just means that, if Beth came to the party, then both Alice and Beth came.

We might write this in a form that is closer to our natural deduction format:

1	Alice came to the party	
2	Beth came to the party	
3	Both Alice and Beth came to the party	&I 1, 2
4	Thus, *if* Beth came to the party, both Alice and Beth came.	→I 2–3

The natural deduction format of lines and indentations is there to replace the words and context like "suppose that". The "suppose that" used on line 2 is represented in the formal system by the indentation and additional line. And like the premises, this line does not need to be justified, it is taken as an assumption, and a line underneath it is drawn. What comes underneath this, on line 3, is still indented, and is within the context of the supposition that Beth came to the party. But once we move to line 4, the additional assumption is no longer in place. It has been *discharged.*

Now let's present this again a little more formally: We started with one premise, 'Alice came to the party', thus:

1	A

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

1	A
2	B

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

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

$$
\begin{array}{r|l|ll}
1 & A & & \\
\hline
2 & & B & \\
\cline{3-3}
3 & & A \,\&\, B & \&\text{I } 1, 2
\end{array}
$$

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

$$
\begin{array}{r|l|ll}
1 & A & & \\
\hline
2 & & B & \\
\cline{3-3}
3 & & A \,\&\, B & \&\text{I } 1, 2 \\
4 & B \rightarrow (A \,\&\, B) & & \rightarrow\text{I } 2\text{–}3
\end{array}
$$

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

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

$$
\begin{array}{r|l|ll}
m & & X & \\
\cline{3-3}
 & & \vdots & \\
n & & Y & \\
 & X \rightarrow Y & & \rightarrow\text{I } m\text{–}n
\end{array}
$$

There can be as many or as few lines as you like between lines m and n.

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

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

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

$$
\begin{array}{r|ll}
1 & P \to Q & \\
2 & Q \to R & \\
\hline
3 & \quad \underline{|\; P} &
\end{array}
$$

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

$$
\begin{array}{r|ll}
1 & P \to Q & \\
2 & Q \to R & \\
\hline
3 & \quad |\; \underline{P} & \\
4 & \quad |\; Q & \to\text{E } 1, 3 \\
5 & \quad |\; R & \to\text{E } 2, 4 \\
6 & P \to R & \to\text{I } 3\text{–}5
\end{array}
$$

The subproof also doesn’t need to start immediately. For example:

$$
\begin{array}{r|ll}
1 & (P \to Q) \,\&\, O & \\
2 & Q \to R & \\
\hline
3 & P \to Q & \&\text{E } 1 \\
4 & \quad |\; \underline{P} & \\
5 & \quad |\; Q & \to\text{E } 3, 4 \\
6 & \quad |\; R & \to\text{E } 2, 5 \\
7 & P \to R & \to\text{I } 4\text{–}6
\end{array}
$$

18.2 Disjunction Elimination

The disjunction elimination rule also makes use of subproofs.

Suppose that Alice came to the party or Beth came to the party. What can you conclude? Not that Alice came to the party; it might be that Beth came to the party instead. Equally, not that Beth came to the party; for it might be that only Alice came. Disjunctions, just by themselves, are hard to work with.

But suppose that we could somehow show both of the following: first, that Alice coming to the party entails that it was fun: second, that Beth coming to the party entails that it was fun. Then if we know that Alice or Beth came to the party, then we know that either way, it was fun. This insight can be expressed in the following rule, which is our disjunction elimination ($\vee$E) rule:

m	$X \vee Y$	
i	$\quad$ | X (assumption)	
	$\quad$ | $\vdots$	
j	$\quad$ | Z	
k	$\quad$ | Y (assumption)	
	$\quad$ | $\vdots$	
l	$\quad$ | Z	
	Z	$\vee$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, $X \vee Y$. Suppose we have two subproofs, showing us that Z follows from the assumption that X, and that Z follows from the assumption that Y. Then we can infer Z itself. As usual, there can be as many lines as you like between i and j, and as many lines as you like between k and l. Moreover, the subproofs and the disjunction can come in any order, and do not have to be adjacent.

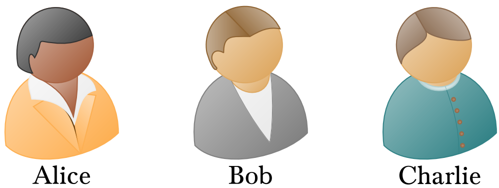
Some examples might help illustrate this. Consider this argument:

$$(P \mathbin{\&} Q) \vee (P \mathbin{\&} R) \therefore P$$

A proof corresponding to this argument is:

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

Consider the following brain teaser:[1]

> Three people are standing in a row looking at each other.
>
> 
>
>
> Alice is happy. Charlie is not happy. Is there someone who is happy who is looking at someone who is not happy?

. . . Think about it!
. . . Answer: Yes. Our Disjunction Elimination rule along with the Law of Excluded Middle allow us to show this. We can demonstrate this in the following argument, which we present in a pseudo-formal style.[2]

[1] Originally by Hector Levesque.

[2] Though, actually, this is most naturally formulated as a validity claim of First Order Logic. We'll walk through the formal proof as formulated in First Order Logic in §34.2.

1	Bob is either happy or he's not happy
2	Suppose Bob is happy
3	Then happy Bob is looking at not-happy Charlie
4	So someone who is happy is looking at someone who is not
5	Suppose Bob is not happy
6	Then happy Alice is looking at not-happy Bob
7	So someone who is happy is looking at someone who is not
8	Therefore, someone who is happy is looking at someone who is not.

Are you convinced now that someone who is happy is looking at someone who is not happy? If not, find a friend and work through it together. Sometimes it can really help to try walking through the argument together.

Coming up with this sort of argument does just take that moment of inspiration to see how this argument will go (that's why it's a brain teaser). This is often the case with arguments that involve the law of excluded middle. We pick it out of nowhere and have to use our inspiration to see how it might be useful. But hopefully, with more examples you'll become familiar with cases where it might be of use. A strategy that might help is: it's a backup option if everything else fails. If it doesn't look like there's any elimination rules to use on your premises or any introduction rules that can get you to your conclusion, then perhaps LEM is the way forwards.

One more example:

$$P \therefore (P \,\&\, D) \vee (P \,\&\, \neg D)$$

Here is a proof corresponding with the argument:

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

18.3 Negation

Negation Introduction

If assuming something leads you to a contradiction, then the assumption must be wrong. This thought motivates the following rule:

m	$\quad X$	
	$\quad \vdots$	
n	$\quad \bot$	
	$\neg X$	$\neg$I m–n

To see this in practice, and interacting with negation, consider this proof:

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

Proof by Contradiction

The next rule is quite similar. If assuming that something is false leads you to a contradiction, then that assumption must be wrong — and so that 'something' must in fact be true.

$$
\begin{array}{r|l l}
m & \quad\begin{array}{|l}\neg X \\ \hline \vdots \\ \bot \end{array} & \\
 & X & \text{PbC } m\text{–}n \\
\end{array}
$$

This is called the method of *proof by contradiction*, or *indirect proof*. It allows us to prove something by assuming its negation and showing that it leads to contradiction. This technique is very common in mathematics.[3]

For example:

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

There is no explicit rule for Negation Elimination. Though both Proof by Contradiction and Contradiction Introduction involve eliminating negations.

18.4 Additional assumptions and subproofs

The rules we have just seen involve the idea of making additional assumptions. These need to be handled with some care.

Consider this proof:

[3]For example it is used to show that $\sqrt{2}$ is irrational. We begin by assuming that $\sqrt{2}$ is rational (i.e. that there is a fraction $\frac{p}{q}$ whose square is 2) and then derive contradiction from this assumption.

$$
\begin{array}{r|l|l}
1 & A & \\
\hline
2 & \quad \begin{array}{|l} B \\ \hline \end{array} & \\
3 & \quad \begin{array}{|l} B \end{array} & \text{R } 2 \\
4 & B \to B & {\to}\text{I } 2\text{–}3
\end{array}
$$

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

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

$$
\begin{array}{r|l|l}
1 & A & \\
\hline
2 & \quad \begin{array}{|l} B \\ \hline \end{array} & \\
3 & \quad \begin{array}{|l} B \end{array} & \text{R } 2 \\
4 & B \to B & {\to}\text{I } 2\text{–}3 \\
5 & B & \text{naughty attempt to invoke } {\to}\text{E } 4, 3
\end{array}
$$

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

We can describe the process of making an additional assumption as one of performing a *subproof*: a subsidiary proof within the main proof. When we start a subproof, we draw another vertical line to indicate that we are no longer in the main proof. Then we write in the assumption upon which the subproof will be based. A subproof can be thought of as essentially posing this question: *what could we show, if we also make this additional assumption?*

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

A subproof is CLOSED when the vertical line for the subproof comes to an end. At that point we say the assumption has been DISCHARGED

We typically do this when we use one of our rules that involve subproofs, such as →I. We introduced the assumption X to allow us to conclude Y; and this reasoning allows us to close the subproof and conclude $X \rightarrow Y$, which no longer relies on the assumption X. Having closed a subproof, we have set aside the additional assumption, so it will be illegitimate to draw upon anything that depends upon that additional assumption. Thus we stipulate:

Any rule whose citation requires mentioning individual lines can mention any earlier lines, *except* for those lines which occur within a closed subproof.
Put another way: you cannot refer back to anything that was obtained using discharged assumptions

This 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.

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

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

1	A	
2	$\quad B$	
3	$\quad\quad C$	
4	$\quad\quad A \& B$	&I 1, 2
5	$\quad C \rightarrow (A \& B)$	→I 3–4
6	$B \rightarrow (C \rightarrow (A \& B))$	→I 2–5

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

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

1	A	
2	$\quad B$	
3	$\quad\quad C$	
4	$\quad\quad A \& B$	&I 1, 2
5	$\quad C \rightarrow (A \& B)$	→I 3–4
6	$B \rightarrow (C \rightarrow (A \& B))$	→I 2–5
7	$C \rightarrow (A \& B)$	naughty attempt to invoke →I 3–4

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

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

Any rule whose citation requires mentioning an entire subproof can mention any earlier subproof, *except* for those subproofs which occur within some *other* closed subproof.

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

It is always permissible to open a subproof with any assumption. However, there is some strategy involved in picking a useful assumption. Starting a subproof with an arbitrary, wacky assumption would just waste lines of the proof. In order to obtain a conditional by →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.

18.5 Proof Strategies

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

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

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

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

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 often 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 X directly, try starting by assuming $\neg X$. If a contradiction follows, then you will be able to obtain X by PbC.

Law of Excluded Middle. If you're hitting a blank, try seeing if there's some instance of LEM that might help you. These arguments do just need some inspiration to see the instance of LEM that'll be helpful.

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

If the argument is actually valid (which is defined using truth-tables) there will be a proof of it somehow…

CHAPTER 19

Proofs and Validity

The system of rules we have set up is not just a game. It helps us understand the validity of arguments.

An argument $X_1, X_2, \ldots, X_n \therefore Y$ may have a proof in the system of natural deduction. Such a proof may look something like:

$$
\begin{array}{r|l}
1 & X_1 \\
2 & X_2 \\
n & \underline{X_n} \\
 & \vdots \\
 & Y
\end{array}
$$

That is, it will start with the premises as assumptions, and proceed following the rules we have given and finishing with the conclusion. It might also have subproofs along the way, something like:

1	X_1	
2	X_2	
n	X_n	
	$\vdots$	
m	Some additional assumption	
	$\vdots$	
m	Maybe some more assumptions	
	$\vdots$	
	$\vdots$	
	$\vdots$	
	Y	

But any subproofs need to have been closed by the time we get to Y. So, for example if we gave

1	A	
2	B	
3	B	R 2

this does not count as a proof corresponding to the argument $A \therefore B$. But

1	A	
2	B	
3	B	R 2
4	$B \to B$	$\to$I 2–3

counts as a proof corresponding to $A \therefore B \to B$ as the subproof has been closed on line 4.

So we have now said when we have a proof in our natural deduction system that corresponds to a particular argument. If we can find a proof then we know that the argument is valid.

If there is a proof in natural deduction corresponding to the argument $X_1 \dots X_n \therefore Y$, then this argument $X_1 \dots X_n \therefore Y$ is valid.

This property of our proof system is called SOUNDNESS. It holds because we only chose rules that matched valid reasoning steps. Recall that $A \rightarrow B, B \therefore A$ is invalid. Had we added a rule such as

	$\vdots$	
n	$X \rightarrow Y$	
	$\vdots$	
m	Y	
	$\vdots$	
	X	Do not do this. Does not follow from n, m

we would then have been able to construct a proof corresponding to the invalid argument $A \rightarrow B, B \therefore A$. We do not have such a rule in our system. All the rules we gave in our system will result in proofs of valid arguments.

We can actually strengthen the link between proofs corresponding to arguments and those argument's validity:

If an argument is valid, then there is a proof of it in natural deduction.

This property of our proof system is called COMPLETENESS.

So for every valid argument there will be some proof. This doesn't mean it is always easy to come up with such a proof, but there will be one. Persist!

CHAPTER 20

Additional rules for TFL

In §17, we introduced the basic rules of our proof system for TFL. In this section, we will add some additional rules to our system. These will make our system much easier to work with. (However, in §21 we will see that they are not strictly speaking *necessary*.)

20.1 Disjunctive syllogism

Here is a very natural argument form.

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

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

m	$X \vee Y$	
n	$\neg X$	
	Y	DS m, n

and

$$
\begin{array}{r|ll}
m & X \vee Y & \\
n & \neg Y & \\
 & X & \text{DS } m, n
\end{array}
$$

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

20.2 Modus tollens

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

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

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

$$
\begin{array}{r|ll}
m & X \rightarrow Y & \\
n & \neg Y & \\
 & \neg X & \text{MT } m, n
\end{array}
$$

As usual, the premises may occur in either order.

20.3 Double-negation

A sentence $\neg\neg X$ is always logically equivalent to X. We can add rules to our system that encode this idea: allowing us to immediately eliminate or introduce double negations:

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

$$
\begin{array}{r|ll}
m & X & \\
 & \neg\neg X & \text{DNI } m
\end{array}
$$

That said, you should be aware that in ordinary language we can sometimes speak in a way that is similar to, but not quite, a double negation. Consider: 'Jane is not *un*happy'. Arguably, one cannot infer 'Jane is happy'. Perhaps the speaker is using this unusual indirect phrasing to draw attention to the possible difference between 'unhappy' and 'not unhappy'. Perhaps what they mean to suggest is that 'Jane is in a state of profound indifference'. Here, then, 'Jane is unhappy' should not be thought of as equivalent to 'It is not the case that Jane is happy', and it should not be symbolised as $\neg H$ but should rather be a separate atomic sentence. So 'Jane is not unhappy' is not then seen as a double negation.

20.4 De Morgan Rules

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

The first De Morgan rule is:

$$
\begin{array}{r|ll}
m & \neg(X \,\&\, Y) & \\
 & \neg X \lor \neg Y & \text{DeM } m
\end{array}
$$

The second De Morgan is the reverse of the first:

$$
\begin{array}{r|ll}
m & \neg X \lor \neg Y & \\
 & \neg(X \,\&\, Y) & \text{DeM } m
\end{array}
$$

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

m	$\neg(X \vee Y)$	
	$\neg X \,\&\, \neg Y$	DeM m

And the fourth is the reverse of the third:

m	$\neg X \,\&\, \neg Y$	
	$\neg(X \vee Y)$	DeM m

There are many more rules one could add to the system as derived rules. But these are all the ones we'll introduce.

CHAPTER 21

Derived rules

In this section, we will see why we introduced the rules of our proof system in two separate batches. In particular, we want to show that the additional rules of §20 are not strictly speaking necessary, but can be derived from the basic rules of §17.

21.1 Derivation of Disjunctive syllogism

Suppose that you are in a proof, and you have something of this form:

$$\begin{array}{r|l} m & X \vee Y \\ n & \neg X \end{array}$$

You now want, on line k, to prove Y. You can do this with the rule of DS, introduced in §20, but equally well, you can do this with the *basic* rules of §17:

m	$X \vee Y$	
n	$\neg X$	
k	$\quad X$	
$k+1$	$\quad \bot$	$\bot$I k, n
$k+2$	$\quad Y$	$\bot$E $k+1$
$k+3$	$\quad Y$	
$k+4$	$\quad Y$	R $k+3$
$k+5$	Y	$\vee$E m, k–$k+2$, $k+3$–$k+4$

To be clear: this is not a proof. Rather, it is a proof *scheme*. (This is why we use letters like m and k to label the lines of the proof rather than numbers.) Whatever sentences of TFL we plugged in for 'X' or 'Y', and whatever lines we were working on, we could produce a bona fide proof. So you can think of this as a recipe for producing proofs.

Indeed, it is a recipe which shows us that, anything we can prove using the rule DS, we can prove (with a few more lines) using just the other rules of §17.

21.2 Derivation of Modus Tollens

Suppose in the course of you proof you already have $X \rightarrow Y$, say on line m, and $\neg Y$ on line n. At some later line, k, you want to get $\neg X$. You can do this with the rule of Modus Tollens (MT), introduced in §20. But you could also do this with the *basic* rules of §17:

m	$X \rightarrow Y$	
n	$\neg Y$	
k	$\quad X$	
$k+1$	$\quad Y$	$\rightarrow$E m, k
$k+2$	$\quad \bot$	$\bot$I $k+1$, n
$k+3$	$\neg X$	$\neg$I k–$k+2$

Again, the rule of MT can be derived from the *basic* rules of §17.

21.3 Derivation of Double-negation rules

Consider the following deduction schema:

$$
\begin{array}{l|lll}
m & X & & \\
j & \quad \neg X & & \\
\cline{2-2}
j+1 & \quad \bot & & \bot\text{I } j, m \\
j+2 & \neg\neg X & & \neg\text{I } j\text{–}(j+1)
\end{array}
$$

and

$$
\begin{array}{l|lll}
m & \neg\neg X & & \\
j & \quad \neg X & & \\
\cline{2-2}
j+1 & \quad \bot & & \bot\text{I } j, m \\
j+2 & X & & \text{PbC } j\text{–}(j+1)
\end{array}
$$

So again, we can derive the double negations rules from the *basic* rules of §17.

21.4 Derivation of De Morgan rules

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

m	$\neg(X \,\&\, Y)$	
j	$X \vee \neg X$	LEM
k	X	
$k+1$	Y	
$k+2$	$X \,\&\, Y$	&I k, $k+1$
$k+3$	$\bot$	$\bot$I $k+2$, m
$k+4$	$\neg Y$	$\neg$I $k+1$–$k+3$
$k+5$	$\neg X \vee \neg Y$	$\vee$I $k+4$
$k+6$	$\neg X$	
$k+7$	$\neg X \vee \neg Y$	$\vee$I $k+6$
$k+8$	$\neg X \vee \neg Y$	$\vee$E j, k–$k+5$, $k+6$–$k+7$

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

m	$\neg X \vee \neg Y$	
k	$X \,\&\, Y$	
$k+1$	X	&E k
$k+2$	Y	&E k
$k+3$	$\neg X$	
$k+4$	$\bot$	$\bot$I $k+1$, $k+3$
$k+5$	$\neg Y$	
$k+6$	$\bot$	$\bot$I $k+2$, $k+5$
$k+7$	$\bot$	$\vee$E m, $k+3$–$k+4$, $k+5$–$k+6$
$k+8$	$\neg(X \,\&\, Y)$	$\neg$I k–$k+7$

Similar demonstrations can be offered explaining how we could derive the third and fourth De Morgan rules. These are left as exercises.

CHAPTER 22

Soundness and completeness

(Non-examinable)A very important result:

> A TFL argument is valid if and only if it can be given a proof in this natural deduction system.

In this chapter, we explain a bit more about how such an argument would go. Soundness is proved in more detail in §44.

22.1 Entails

For this chapter we will make use of a further symbols:

We use the symbol $\vDash$ as shorthand for 'entails'. Rather than saying that the TFL sentences $X_1, X_2, \ldots$ and X_n together entail Y, we will abbreviate this by:

$$X_1, X_2, \ldots, X_n \vDash Y$$

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

Let me be clear. '$\vDash$' is not a symbol of TFL. Rather, it is a symbol of our metalanguage, augmented English (recall the difference between object language and metalanguage from §8). So the metalanguage sentence:

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

is just an abbreviation for the English sentence:

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

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

$$\vDash Y$$

22.2 Proves

The following expression:

$$X_1, X_2, \ldots, X_n \vdash Y$$

means that there is some proof which starts with assumptions among $X_1, X_2, \ldots, X_n$ and ends with Y (and contains no undischarged assumptions other than those we started with). Derivatively, we will write:

$$\vdash X$$

to mean that there is a proof of X with no assumptions.

The symbol '$\vdash$' is called the *single turnstile*. We want to emphasize that this is not the double turnstile symbol ('$\vDash$') that we introduced to symbolize entailment. The single turnstile, '$\vdash$', concerns the existence of proofs; the double turnstile, '$\vDash$', concerns the existence of valuations (or interpretations, when used for FOL). *They are very different notions.*

22.3 Their equivalence

However, it turns out that they are equivalent. That is:

$$X_1, X_2, \ldots, X_n \vdash Y$$
if and only if
$$X_1, X_2, \ldots, X_n \vDash Y$$

A full proof here goes well beyond the scope of this book. However, we can sketch what it would be like.

Soundness

This argument from $\vdash$ to $\vDash$ is the problem of SOUNDNESS. A proof system is SOUND if there are no derivations of arguments that can be shown invalid by truth tables. Demonstrating that the proof system is sound would require showing that *any* possible proof is the proof of a valid argument. It would not be enough simply to succeed when trying to prove many valid arguments and to fail when trying to prove invalid ones.

The proof that we will sketch depends on the fact that we initially defined a sentence of TFL using a recursive definition (see p. 39). We could have also used recursive definitions to define a proper proof in TFL and a proper truth table. (Although we didn't.) If we had these definitions, we could then use a *recursive proof* to show the soundness of TFL. A recursive proof works the same way as a recursive definition. With the recursive definition, we identified a group of base elements that were stipulated to be examples of the thing we were trying to define. In the case of a TFL sentence, the base class was the set of sentence letters A, B, C, ... We just announced that these were sentences. The second step of a recursive definition is to say that anything that is built up from your base class using certain rules also counts as an example of the thing you are defining. In the case of a definition of a sentence, the rules corresponded to the five sentential connectives (see p. 39). Once you have established a recursive definition, you can use that definition to show that all the members of the class you have defined have a certain property. You simply prove that the property is true of the members of the base class, and then you prove that the rules for extending the base class don't change the property. This is what it means to give a recursive proof.

Even though we don't have a recursive definition of a proof in TFL, we can sketch how a recursive proof of the soundness of TFL would go. Imagine a base class of one-line proofs, one for each of our basic rules of inference. The members of this class would look like this $X, Y \vdash X \& Y$; $X \& Y \vdash X$; $X \vee Y, \neg X \vdash Y$ ... etc. Since some rules have a couple different forms, we would have to have add some members to this base class, for instance $X \& Y \vdash Y$ Notice that these are all statements in the metalanguage. The proof that TFL is sound is not a part of TFL, because TFL does not have the power to talk about itself.

You can use truth tables to prove to yourself that each of these one-line proofs in this base class is valid$_{\vDash}$. For instance the proof $X, Y \vdash X \& Y$ corresponds to a truth table that shows $X, Y \vDash X \& Y$ This establishes the first part of our recursive proof.

The next step is to show that adding lines to any proof will never change a valid$_{\vDash}$ proof into an invalid$_{\vDash}$ one. We would need to do this for each of our basic rules of inference. So, for instance, for &I we need to show that for any proof $X_1, \ldots, X_n \vdash Y$ adding a line where we use &I to infer $Z \,\&\, V$, where $Z \,\&\, V$ can be legitimately inferred from $X_1, \ldots,$ X_n, Y, would not change a valid proof into an invalid proof. But wait, if we can legitimately derive $Z \,\&\, V$ from these premises, then Z and V must be already available in the proof. They are either already among $X_1, \ldots, X_n$, B, or can be legitimately derived from them. As such, any truth table line in which the premises are true must be a truth table line in which Z and V are true. According to the characteristic truth table for &, this means that $Z \,\&\, V$ is also true on that line. Therefore, $Z \,\&\, V$ validly follows from the premises. This means that using the &E rule to extend a valid proof produces another valid proof.

In order to show that the proof system is sound, we would need to show this for the other inference rules. Since the derived rules are consequences of the basic rules, it would suffice to provide similar arguments for the other basic rules. This tedious exercise falls beyond the scope of this book.

So we have shown that $X \vdash Y$ implies $X \vDash Y$. What about the other direction, that is why think that *every* argument that can be shown valid using truth tables can also be proven using a derivation.

Completeness

This is the problem of completeness. A proof system has the property of COMPLETENESS if and only if there is a derivation of every semantically valid argument. Proving that a system is complete is generally harder than proving that it is sound. Proving that a system is sound amounts to showing that all of the rules of your proof system work the way they are supposed to. Showing that a system is complete means showing that you have included *all* the rules you need, that you haven't left any out. Showing this is beyond the scope of this book. The important point is that, happily, the proof system for TFL is both sound and complete. This is not the case for all proof systems or all formal languages. Because it is true of TFL, we can choose to give proofs or give truth tables—whichever is easier for the task at hand.

22.4 Other Semantic and Proof Theoretic Notions

Now we know that the proof theoretic and truth-table methods are equivalent, we can use them interchangable depending on which is more useful.

We can also use proof theoretic methods for determining other logical notions, such as being consistent. We summarise how one would define them in 22.1

In fact, we can give general guidelines about when it's best to give proofs and when it is best to give truth tables. We do this in 22.2:

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

Table 22.1: Two ways to define logical concepts.

Logical property	**To prove it present**	**To prove it absent**
Being a tautology	Derive the sentence	Find the false line in the truth table for the sentence
Being a contradiction	Derive the negation of the sentence	Find the true line in the truth table for the sentence
Contingency	Find a false line and a true line in the truth table for the sentence	Prove the sentence or its negation
Equivalence	Derive each sentence from the other	Find a line in the truth tables for the sentence where they have different values
Consistency	Find a line in truth table for the sentence where they all are true	Derive a contradiction from the sentences
Validity	Derive the conclusion from the premises	Find no line in the truth table where the premises are true and the conclusion false.

Table 22.2: When to provide a truth table and when to provide a proof.

CHAPTER 23

Proof-theoretic concepts

(Non-examinable)

Armed with our '$\vdash$' symbol, we can introduce some new terminology.

> X is a THEOREM iff $\vdash X$

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

1	$\quad\mid\; A \,\&\, \neg A$	
2	$\quad\mid\; A$	&E 1
3	$\quad\mid\; \neg A$	&E 1
4	$\quad\mid\; \bot$	$\bot$I 2, 3
5	$(A \,\&\, \neg A) \to \bot$	$\to$I 1–4
6	$\neg(A \,\&\, \neg A)$	$\neg$I 5

We have therefore proved '$\neg(A \,\&\, \neg A)$' on no (undischarged) assumptions. This particular theorem is an instance of what is sometimes called *the Law of Non-Contradiction*.

To show that something is a theorem, you just have to find a suitable proof. It is typically much harder to show that something is *not* a theorem. To do this, you would have to demonstrate, not just that certain proof strategies fail, but that *no* proof is possible. Even if you fail in trying to prove a sentence in a thousand different ways, perhaps the proof is just too long and complex for you to make out. Perhaps you just didn't try hard enough.

Here is another new bit of terminology:

> Two sentences X and Y are PROVABLY EQUIVALENT iff each can be proved from the other; i.e., both $X \vdash Y$ and $Y \vdash X$.

As in the case of showing that a sentence is a theorem, it is relatively easy to show that two sentences are provably equivalent: it just requires a pair of proofs. Showing that sentences are *not* provably equivalent would be much harder: it is just as hard as showing that a sentence is not a theorem.

Here is a third, related, bit of terminology:

> The sentences $X_1, X_2, \ldots, X_n$ are PROVABLY INCONSISTENT iff a contradiction can be proved from them, i.e. $X_1, X_2, \ldots, X_n \vdash \bot$. If they are not INCONSISTENT, we call them PROVABLY CONSISTENT.

It is easy to show that some sentences are provably inconsistent: you just need to prove a contradiction from assuming all the sentences. Showing that some sentences are not provably inconsistent is much harder. It would require more than just providing a proof or two; it would require showing that no proof of a certain kind is *possible*.

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

	Yes	**No**
theorem?	one proof	all possible proofs
inconsistent?	one proof	all possible proofs
equivalent?	two proofs	all possible proofs
consistent?	all possible proofs	one proof

PART V

First-order logic

CHAPTER 24

Building blocks of FOL

24.1 The need to decompose sentences

We have been studying arguments, and in particular their validity. In section 12.2 we gave a strategy for checking the validity of an argument by using TFL. That was:

1. Find the structure of the argument.
   Identify the premises and conclusion.
2. Symbolise the argument in TFL.
3. Check if the TFL argument is valid.
   - ▹ Using truth tables to look for a valuation providing a counter example. If there is no such valuation, then it is valid.
   - ▹ Or, use natural deduction to show that it is valid.

However, this only allows you to conclude that the original English language argument is valid. But what if the best TFL symbolisation is invalid? Consider the following arguments:

1. • Alice is a logician.
   • All logicians wear funny hats.

- Therefore: Alice wears a funny hat.

2. - Everyone who loves Manchester United hates Manchester City.
   - Manchester City is not hated by everyone.
   - Therefore: there is at least one person who doesn't love Manchester United.

We can symbolise these in TFL (follow the strategy as in 43). Since we cannot paraphrase any of these sentences with 'and', 'if', 'or' or 'not', we simply have to use atomic sentences. We thus offer the symbolisation:

$$L, A \therefore H$$

with the symbolisation

L: Alice is a logician.
A: All logicians wear funny hats.
H: Alice wears a funny hat.

And for the second argument we would symbolise this as:

$$P, \neg Q \quad \therefore R$$

using

P: Everyone who loves Manchester United hates Manchester City.
Q: Manchester City is hated by everyone.
R: There is at least one person who doesn't love Manchester United.

Both of these TFL arguments are invalid. But the original English arguments are themselves valid.

The problem is not that we have made a mistake while symbolizing the argument. The problem lies with TFL itself. The expressive power of TFL is not rich enough to explain why these English arguments are valid. TFL can recognise arguments that are valid in virtue of the meanings of 'and', 'if' etc. But these arguments are valid in virtue of something else.

The first argument is valid in virtue of the meaning of 'all' and the fact that 'Alice' is a name. The second argument is valid in virtue of the meanings of 'there is', 'every' and 'not'.

We will introduce a new logical language which allows us to capture this. We will call this language *first-order logic*, or *FOL*.

The details of FOL will be explained throughout this chapter.

24.2 Names and Predicates

Consider

> Alice is a logician.

In TFL we used an atomic sentence to represent this. In FOL we will break it into two components: a name and a predicate.

$$\overbrace{\text{Alice}}^{\text{Name}}\ \overbrace{\text{is a logician.}}^{\text{Predicate}}$$

A name picks out an individual. The name 'Alice' is picking out some particular person, Alice.

A predicate expresses a property, in this case the property of being a logician. The predicate is:

> _____ is a logician

In First Order Logic, FOL, we can symbolise these different components. We will use lower-case letters like $a, b, c \ldots$ for names (except x, y, z which are used for variables as we will later see), and upper case letters like $A, B, C, \ldots$ for predicates (except X, Y, Z, which are used for metavariables). We can also add numbered subscripts if needed, for example using d_{27} as a name, or H_{386} as a predicate.

Like in TFL, when symbolising we have to give a symbolisation key to specify how to interpret the predicates and names. In this case, we might give:

a: Alice
Lx: $\underline{\qquad}_x$ is a logician

and we can then symbolise 'Alice is a logician' as

$$La.$$

(We will say more about the "x" subscript later.)

Note that in FOL the name follows the predicate: we have to write it as La. The property of being a logician applies to Alice.

As in TFL our choice of which letter to use for our name or predicate doesn't matter. It would be equally good to give

a: Alice
Px: $\underline{\qquad}_x$ is a logician

And then symbolise 'Alice is a logician' as

$$Pa.$$

Let's see some other example sentences which have this same form. Each of these sentences could similarly be symbolised as Pa, though the symbolisation key would have to change in each of these instances.

1. Rocky is strong
2. Joe Biden is a Democrat
3. Michael Palin is a member of Monty Python

In each of these cases the relevant symbolisation key would then be:

1
a: Rocky
Px: $\_\_\_\_\__x$ is strong

2
a: Joe Biden
Px: $\_\_\_\_\__x$ is a Democrat

3
a: Michael Palin
Px: $\_\_\_\_\__x$ is a member of Monty Python

Names don't have to name people, for example we can also symbolise

4. The Tower of London is in England.

as Pa using the symbolisation key:

a: The Tower of London
Px: $\_\_\_\_\__x$ is in England

What is important, though, is that what we are symbolising as a name in FOL refers to a *specific* person, place, or thing.

Consider

5. Buses are red.

You might think that this has the same form and symbolise it as La with the symbolisation key:

a: Buses
Lx: $\_\_\_\_\__x$ is red

But this would be wrong. Do not do this. The reason is that 'Buses' does not refer to a specific thing, it refers to a great many objects.

24.3 Names, predicates and connectives

In FOL we will also make use of all of the tools from TFL. We can symbolise

6. $\underbrace{\underbrace{\text{Joe is happy}}_{Hj} \text{ and } \underbrace{\text{Katie is sad}}_{Sk}}_{(Hj\&Sk)}$.

as

$$Hj \,\&\, Sk$$

with the symbolisation key:

Hx: $\_\_\_\_\__x$ is happy
Sx: $\_\_\_\_\__x$ is sad
j: Joe
k: Katie

To symbolise

7. Joe and Katie are happy

we observe that it can be naturally paraphrased as 'Joe is happy and Katie is happy' and thus symbolised as

$$Hj \,\&\, Hk$$

To symbolise

8. If Joe is happy, then Katie is too

we observe that it can be naturally paraphrased as 'If Joe is happy then Katie is happy' and thus symbolised as

$$Hj \rightarrow Hk$$

We can also symbolise more complex sentences, for example:

9. If Joe is not happy then Katie or Billy is sad.

$$\neg Hj \rightarrow (Sk \vee Sb)$$

One final example. To symbolise:

10. Herbie is a red car

we might simply offer

$$Ah$$

using

Ax: $\_\_\_\_\__x$ is a red car
h: Herbie

But it is more informative to observe that we can naturally paraphrase it as 'Herbie is red and Herbie is a car' so symbolise it as

$$Rh \,\&\, Ch$$

using

Rx: $\_\_\_\_\__x$ is red
Cx: $\_\_\_\_\__x$ is a car
h: Herbie

Since this latter symbolisation extracts more of the information from the original sentence, it is generally going to be better.

24.4 Many-placed predicates

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

However, other predicates concern the *relation* between two things. Here are some examples of relational predicates in English:

_____ loves _____
_____ is to the left of _____
_____ is in debt to _____

These are TWO-PLACE predicates. They need to be filled in with two terms in order to make a sentence. They express a relationship between two objects.

Now there is a little foible with the above. We have used the same symbol, '_____', to indicate a gap formed by deleting a term from a sentence. However (as Frege emphasized), these are *different* gaps. To obtain a sentence, we can fill them in with the same term, but we can equally fill them in with different terms, and in various different orders. The following are all perfectly good sentences, and they all mean very different things:

Karl loves Karl
Karl loves Imre
Imre loves Karl
Imre loves Imre

The point is that we need to keep track of the gaps in predicates, so that we can keep track of how we are filling them in.

To keep track of the gaps, we will label them. The labelling conventions we will adopt are best explained by example. Suppose we want to symbolize the following sentences:

11. Karl loves Imre.
12. Imre loves himself.
13. Karl loves Imre, but not vice versa.
14. Karl is loved by Imre.

We will start with the following symbolisation key:

domain: people
i: Imre
k: Karl
Lxy: $\_\_\_\_\__x$ loves $\_\_\_\_\__y$

Sentence 11 will now be symbolized by Lki.

Sentence 12 can be paraphrased as 'Imre loves Imre'. It can now be symbolized by Lii.

Sentence 13 is a conjunction. We might paraphrase it as 'Karl loves Imre, and Imre does not love Karl'. It can now be symbolized by $Lki \,\&\, \neg Lik$.

Sentence 14 might be paraphrased by 'Imre loves Karl'. It can then be symbolized by Lik. Of course, this slurs over the difference in tone between the active and passive voice; such nuances are lost in FOL.

This last example, though, highlights something important. Suppose we add to our symbolization key the following:

Mxy: $\_\_\_\_\__y$ loves $\_\_\_\_\__x$

Here, we have used the same English word ('loves') as we used in our symbolization key for Lxy. However, we have swapped the order of the *gaps* around (just look closely at those little subscripts!) So Mki

and Lik now *both* symbolize 'Imre loves Karl'. Mik and Lki now *both* symbolize 'Karl loves Imre'. Since love can be unrequited, these are very different claims.

The moral is simple. When we are dealing with predicates with more than one place, we need to pay careful attention to the order of the places.

Predicates can have more than two places.

For example, consider

15. David bought the necklace for Victoria.

We symbolise this as

$$Bdna$$

using the symbolisation key:

d: David
n: the necklace
a: Victoria
$Rxyz$: $\_\_\_\_\__x$ bought $\_\_\_\_\__y$ for $\_\_\_\_\__z$

There is no limit to the number of places that a predicate may have.

16. The daughter of Gregor and Hilary is a friend of the first daughter of Bill and Michelle.

We symbolise this as

$$Rabcd$$

using:

a: Gregor
b: Hilary
c: Bill
d: Michelle
$Rx_1x_2x_3x_4$: The daughter of $\_\_\_\_\__{x_1}$ and $\_\_\_\_\__{x_2}$ is a friend of the first daughter of $\_\_\_\_\__{x_3}$ and $\_\_\_\_\__{x_4}$.

24.5 Universal Quantifier

Consider

17. Everyone wears a funny hat

This doesn't say of any specific individual that they wear a funny hat, but it says everyone does so. To express this, we introduce the ∀ symbol. This is called the *universal quantifier.*

We read ∀ as "for all" or "for every". In this case what do we want to say holds of all the people? We want to say that they wear a funny hat. In this sentence we used the "they". This doesn't refer to any particular person, Harry or Katie, instead it can refer to anyone. That is, we are using it as a variable. We might then paraphrase "Everyone wears a funny hat" more explicitly as:

For everyone $_x$: x wears a funny hat.

Here we have made explicit the variable as x. In FOL we can also use y, z, or also, for example, x_{32} as variables. Quantifiers always have to be followed immediately by a variable.

If we wanted to symbolise "Alice wears a funny hat" we would use Fa. To symbolise "Everyone wears a funny hat", we paraphrase it as "For everyone $_x$: x wears a funny hat." and then symbolise it as $\forall x Fx$.

Whatever we wanted to say of an individual we can now say of everyone using this quantifier. Consider

18. Everyone is happy and wears a funny hat

We can break this up:

$$\underbrace{\underbrace{\text{For everyone }_x}_{\forall x} : \underbrace{\underbrace{x\text{ is happy}}_{Hx} \text{ and } \underbrace{x\text{ wears a funny hat}}_{Fx}}_{(Hx \& Fx)}}_{\forall x(Hx \& Fx)}.$$

So we can symbolise it as

$$\forall x(Hx \& Fx)$$

We have here been using $\forall x$ to be read out-loud as "for everyone". But how the quantifier should be read depends on the DOMAIN. The domain is the collection of things that we are talking about. $\forall x$ should be read as "for all objects in the domain $_x$". If the domain also contains dogs, or landmarks, then it also says something about those dogs, or landmarks. We say that the quantifiers *range over* the objects in the domain.

If I give

Ex: $\_\_\_\_\__x$ is energetic
domain: dogs

Then $\forall x Ex$ symbolises "All dogs are energetic". If the domain is all dogs, then we'd then read $\forall x$ as "For every dog $_x$: ...".

If I have a domain consisting of landmarks, then $\forall x$ is read as "For every landmark $_x$: ...".

Domains are useful even when we are just talking about people. When we use sentences like "Everyone wears a funny hat" in English, we usually do not mean everyone now alive on the Earth. We certainly do not mean everyone who was ever alive or who will ever live. We usually mean something more modest: everyone now in the building, everyone enrolled in the ballet class, or whatever.

The domain can be chosen however you like, however, in FOL domains have to contain at least one object.

24.6 Existential Quantifier

The Universal Quantifier, $\forall$, allows us to capture English notions like "every", "for all" and "any". The final component of FOL is the Existential Quantifier, $\exists$. This allows us to capture "for some", "there exists".

To symbolise

19. Someone is angry.

We paraphrase it as:

- $\underbrace{\text{There is someone }_x\text{ such that:}}_{\exists x}$ $\underbrace{x\text{ is angry.}}_{Ax}$

and symbolise it as $\exists x Ax$ giving the symbolisation key

domain: people
Ax: $\_\_\_\_\__x$ is angry

To symbolise

20. There is a logician who wears glasses

$$\underbrace{\underbrace{\text{There is someone}_x \text{ such that}}_{\exists x} : \underbrace{\underbrace{x \text{ is a logician}}_{Lx} \text{ and } \underbrace{x \text{ wears glasses}}_{Gx}}_{(Lx\&Gx)}.}_{\exists x(Lx\&Gx)}$$

giving our symbolisation key:

domain: people
Lx: ______$_x$ is a logician
Gx: ______$_x$ wears glasses

To symbolise

21. There is a Polish woman who won the Nobel Prize

We break this up as:

$$\underbrace{\underbrace{\text{There is someone}_x}_{\exists x} : \underbrace{\underbrace{\underbrace{x \text{ is a Polish woman}}_{(\underbrace{x \text{ is Polish}}_{Px} \text{ and } \underbrace{x \text{ is a woman}}_{Wx})}}_{(Px\&Wx)} \text{ and } \underbrace{x \text{ won the Nobel Prize}}_{Nx}}_{((Px\&Wx)\&Nx)}}_{\exists x((Px\&Wx)\&Nx)}$$

So we symbolise it as:

$$\exists x((Px \,\&\, Wx) \,\&\, Nx)$$

giving our symbolisation key:

domain: people
Px: ______$_x$ is polish
Wx: ______$_x$ is a woman
Nx: ______$_x$ won the Nobel Prize

As for the universal quantifier, how to read "$\exists x$" depends on the domain. We might talk not about people but about dogs. If our domain is dogs, then we read $\exists x$ as "There is a dog $_x$ such that:".

If we want to symbolise:

22. Some dog is badly behaved.

We can use

$$\underbrace{\underbrace{\text{There is a dog }_x\text{ such that}}_{\exists x} : \underbrace{x\text{ is badly behaved}}_{Bx}}_{\exists x Bx}$$

giving our symbolisation key:

domain: dogs
Bx: ______$_x$ is badly behaved

Before going further with more symbolisations and symbolisations involving many-placed predicates

24.7 Symbolisations

Before moving to symbolise more complex sentences, we explicitly summarise our strategy for symbolising complex sentences. This extends the strategy that we used for TFL in §6:

1. See if the sentence can be paraphrased in English in one of the standard forms.
   - ▹ If not, it's an atomic formula: identify the predicate and the variables or names.
2. Use the symbolisation trick for that form.
3. Repeat the procedure with the components. Etc.

Our key forms are:

English paraphrase	**Symbolisation**
Everything (in the domain) $_x$ is such that:	$\forall x \ldots$
Something (in the domain) $_x$ is such that:	$\exists x \ldots$
It is not the case that X	$\neg X$
X and Y	$(X \mathbin{\&} Y)$
X or Y	$(X \vee Y)$
If X, then Y	$(X \rightarrow Y)$
X if and only if Y	$(X \leftrightarrow Y)$

Also remember that there were various further tricks from II, such as 'X only if Y' as $(X \to Y)$ and 'Unless X, Y' as $(X \lor Y)$. These still apply in the FOL setting. We will also see some more such tricks later.

24.8 Clarification on Domains

In FOL, the domain must always include at least one thing. Moreover, in English we can infer 'something is angry' from 'Gregor is angry'. In FOL, then, we will want to be able to infer $\exists x Ax$ from Ag. So we will insist that each name must pick out exactly one thing in the domain. If we want to name people in places beside Chicago, then we need to include those people in the domain.

> A domain must have *at least* one member. A name must pick out *exactly* one member of the domain, but a member of the domain may be picked out by one name, many names, or none at all.

Non-referring terms (Further philosophical interest)

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

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

23. Chimera is angry.
24. Chimera is not angry.

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

domain: creatures on Earth
Ax: $\underline{\quad\quad}_x$ is angry.
c: chimera

We could then symbolize sentence 23 as Ac and sentence 24 as $\neg Ac$.

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

One option is to say that sentence 23 is not true, because there is no chimera. If sentence 23 is false because it talks about a non-existent thing, then sentence 24 is false for the same reason. Yet this would mean that Ac and $\neg Ac$ would both be false. Given the truth conditions for negation, this cannot be the case.

Since we cannot say that they are both false, what should we do? Another option is to say that sentence 23 is *meaningless* because it talks about a non-existent thing. So Ac would be a meaningful expression in FOL for some interpretations but not for others. Yet this would make our formal language hostage to particular interpretations. Since we are interested in logical form, we want to consider the logical force of a sentence like Ac apart from any particular interpretation. If Ac were sometimes meaningful and sometimes meaningless, we could not do that.

This is the *problem of non-referring terms*, and we will return to it later (see p. 244.) The important point for now is that each name of FOL *must* refer to something in the domain, although the domain can contain any things we like. If we want to symbolize arguments about mythological creatures, then we must define a domain that includes them. This option is important if we want to consider the logic of stories. We can symbolize a sentence like 'Sherlock Holmes lived at 221B Baker Street' by including fictional characters like Sherlock Holmes in our domain.

24.9 Symbolisation with Many-Placed Predicates

To symbolise

25. Everyone loves Alice.

We want to paraphrase it in one of our standard forms, which we do as:

$$\underbrace{\text{For everyone }_x}_{\forall x} : \underbrace{x \text{ loves Alice.}}_{Lxa}$$

So we give the symbolisation

$$\forall x Lxa$$

with the symbolisation key:

domain: people

Lxy: $\underline{\hspace{2em}}_x$ loves $\underline{\hspace{2em}}_y$

a: Alice

If we instead want to symbolise

26. Alice loves everyone.

We paraphrase this as:

$$\underbrace{\text{For everyone } _x}_{\forall x} : \underbrace{\text{Alice loves } x.}_{Lax}$$

So we give the symbolisation

$$\forall x Lax$$

To symbolise

27. Someone loves themselves.

We paraphrase this as:

$$\underbrace{\text{For someone } _x}_{\exists x} : \underbrace{x \text{ loves } x.}_{Lxx}$$

So we give the symbolisation

$$\exists x Lxx$$

If we want to symbolise

28. Some dog likes playing with Finley.

We can do:

$$\underbrace{\text{For some dog } _x}_{\exists x} : \underbrace{x \text{ likes playing with Finley.}}_{Pxf}$$

So we'd offer $\exists x\, Pxf$ with the symbolisation key:

domain: dogs
Pxy: $\_\_\_\_\__x$ likes playing with $\_\_\_\_\__y$
f: Finley

This symbolisation is only legitimate, though, if Finley is referring to a dog rather than, for example, a person. This is because, as we said in 24.8, names have to name members of the domain. If Finley is a person then we have to ensure that our domain contains people too. But then how do we symbolise “for some dog”?

We can instead paraphrase it as:

$$\underbrace{\underbrace{\text{For some thing } x}_{\exists x} : \underbrace{\underbrace{x \text{ is a dog}}_{Dx} \text{ and } \underbrace{x \text{ likes playing with Finley.}}_{Pxf}}_{(Dx\&Pxf)}}_{\exists x(Dx\&Pxf)}$$

24.10 Quantifiers inside a sentence

All the sentences we’ve considered so far have the quantifiers at the beginning of the sentence. But we can also use truth functional connectives to combine sentences of FOL.

29. Finley is not quiet, but some dog is.

We work as follows:

$$\underbrace{\underbrace{\text{Finley is not quiet}}_{\underbrace{\text{it is not the case that } \underbrace{\text{Finley is quiet}}_{Qf}}_{\neg Wf}} \quad \text{and} \quad \underbrace{\text{some dog is quiet.}}_{\underbrace{\underbrace{\text{there is some dog } x \text{ such that}}_{\exists x} \underbrace{x \text{ is quiet}}_{Qx}}_{\exists xQx}}}_{(\neg Qf\&\exists xQx)}$$

So we symbolise this sentence as

$$(\neg Qf \,\&\, \exists xQx)$$

giving the symbolisation key

domain: dogs
Qx: $\rule{2em}{0.4pt}_x$ is quiet
f: Finley

Note, that as per ??, this symbolisation is only legitimate assuming that Finley names a dog. Names have to name members of the domain.

Consider:

30. Not every dog is quiet

We work as follows:

$$\underbrace{\underbrace{\text{It is not the case that}}_{\neg} : \underbrace{\text{every dog is quiet}}_{\underbrace{\underbrace{\text{for every dog}_{x}}_{\forall x} : \underbrace{x \text{ is quiet}}_{Qx}}_{\forall x Qx}}}_{\neg \forall x Qx}$$

So we symbolise this sentence as

$$\neg \forall x Qx$$

We now have the tools to symbolise our second argument from the introduction.

- Everyone who loves Manchester United hates Manchester City.
- Manchester City is not hated by everyone.
- Therefore: there is at least one person who doesn't love Manchester United.

CHAPTER 25

Common Quantifier Phrases and Domains

25.1 Common quantifier phrases

Consider these sentences:

1. Every coin in my pocket is a quarter.
2. Some coin on the table is a dime.
3. Not all the coins on the table are dimes.
4. None of the coins in my pocket are dimes.

In providing a symbolization key, we need to specify a domain. Since we are talking about coins in my pocket and on the table, the domain must at least contain all of those coins. Since we are not talking about anything besides coins, we let the domain be all coins. Since we are not talking about any specific coins, we do not need to deal with any names. So here is our key:

domain: all coins

Px: ______$_x$ is in my pocket
Tx: ______$_x$ is on the table
Qx: ______$_x$ is a quarter
Dx: ______$_x$ is a dime

Sentence 1 is most naturally symbolized using a universal quantifier. The universal quantifier says something about everything in the domain, not just about the coins in my pocket. Sentence 1 can be paraphrased as 'for any coin, *if* that coin is in my pocket *then* it is a quarter'. So we can symbolize it as $\forall x(Px \rightarrow Qx)$.

Since sentence 1 is about coins that are both in my pocket *and* that are quarters, it might be tempting to symbolize it using a conjunction. However, the sentence $\forall x(Px \,\&\, Qx)$ would symbolize the sentence 'every coin is both a quarter and in my pocket'. This obviously means something very different than sentence 1. And so we see:

> If a sentence can be paraphrased in English as
>
> 'every F is G',
> 'all Fs are Gs', or
> 'any F is a G',
>
> it can be symbolised as can be symbolized as
>
> $$\forall x(Fx \rightarrow Gx).$$

Sentence 2 is most naturally symbolized using an existential quantifier. It can be paraphrased as 'there is some coin which is both on the table and which is a dime'. So we can symbolize it as $\exists x(Tx \,\&\, Dx)$.

Notice that we needed to use a conditional with the universal quantifier, but we used a conjunction with the existential quantifier. Suppose we had instead written $\exists x(Tx \rightarrow Dx)$. That would mean that there is some object in the domain of which $(Tx \rightarrow Dx)$ is true. Recall that, in TFL, $X \rightarrow Y$ is logically equivalent (in TFL) to $\neg X \vee Y$. This equivalence will also hold in FOL. So $\exists x(Tx \rightarrow Dx)$ is true if there is some object in the domain, such that $(\neg Tx \vee Dx)$ is true of that object. That is, $\exists x(Tx \rightarrow Dx)$ is true if some coin is *either* not on the table *or* is a dime. Of course there is a coin that is not on the table: there are coins lots of other places. So it is *very easy* for $\exists x(Tx \rightarrow Dx)$ to be true. A conditional will usually be the natural connective to use with a universal quantifier, but a conditional within the scope of an existential quantifier tends to say something very weak indeed. As a general rule of thumb, do not put conditionals in the scope of existential quantifiers unless you are sure that you need one.

> If a sentence can be paraphrased in English as
>
> 'some F is G',
> 'there is some F that is G',
> 'some F is G', or
> 'there is at least on F that is a G'
>
> it can be symbolised as can be symbolized as
>
> $$\exists x(Fx \,\&\, Gx).$$

Sentence 3 can be paraphrased as, 'It is not the case that every coin on the table is a dime'. So we can symbolize it by $\neg\forall x(Tx \rightarrow Dx)$. You might look at sentence 3 and paraphrase it instead as, 'Some coin on the table is not a dime'. You would then symbolize it by $\exists x(Tx \,\&\, \neg Dx)$. Although it is probably not immediately obvious yet, these two sentences are logically equivalent. (This is due to the logical equivalence between $\neg\forall x X$ and $\exists x\neg X$, mentioned in §24, along with the equivalence between $\neg(X \rightarrow Y)$ and $X \,\&\, \neg Y$.)

> If a sentence can be paraphrased in English as
>
> 'not all Fs are Gs',
>
> it can be symbolised as can be symbolized as
>
> $\neg\forall x(Fx \rightarrow Gx)$, or
> $\exists x(Fx \,\&\, \neg Gx)$.

Sentence 4 can be paraphrased as, 'It is not the case that there is some dime in my pocket'. This can be symbolized by $\neg\exists x(Px \,\&\, Dx)$. It might also be paraphrased as, 'Everything in my pocket is a non-dime', and then could be symbolized by $\forall x(Px \rightarrow \neg Dx)$. Again the two symbolizations are logically equivalent; both are correct symbolizations of sentence 4.

If a sentence can be paraphrased in English as

'no Fs are Gs',

it can be symbolised as can be symbolized as

$$\neg\exists x(Fx \,\&\, Gx), \text{ or}$$
$$\forall x(Fx \rightarrow \neg Gx).$$

Finally, consider 'only', as in:

5. Only dimes are on the table.

How should we symbolize this? A good strategy is to consider when the sentence would be false. If we are saying that only dimes are on the table, we are excluding all the cases where something on the table is a non-dime. So we can symbolize the sentence the same way we would symbolize 'No non-dimes are on the table.' Remembering the lesson we just learned, and symbolizing 'x is a non-dime' as '$\neg Dx$', the possible symbolizations are: '$\neg\exists x(Tx \,\&\, \neg Dx)$', or alternatively: '$\forall x(Tx \rightarrow \neg\neg Dx)$'. Since double negations cancel out, the second is just as good as '$\forall x(Tx \rightarrow Dx)$'. In other words, 'Only dimes are on the table' and 'Everything on the table is a dime' are symbolized the same way.

If a sentence can be paraphrased in English as

'only Fs are Gs',

it can be symbolised as can be symbolized as

$$\neg\exists x(Gx \,\&\, \neg Fx), \text{ or}$$
$$\forall x(Gx \rightarrow Fx)$$

25.2 Empty predicates

In §24, we emphasized that a name must pick out exactly one object in the domain. However, a predicate need not apply to anything in the domain. A predicate that applies to nothing in the domain is called an EMPTY PREDICATE. This is worth exploring.

Suppose we want to symbolize these two sentences:

6. Every monkey knows sign language
7. Some monkey knows sign language

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

domain: animals
Mx: $\_\_\_\_\_\__x$ is a monkey.
Sx: $\_\_\_\_\_\__x$ knows sign language.

Sentence 6 can now be symbolized by $\forall x(Mx \rightarrow Sx)$. Sentence 7 can be symbolized as $\exists x(Mx \,\&\, Sx)$.

It is tempting to say that sentence 6 *entails* sentence 7. That is, we might think that it is impossible for it to be the case that every monkey knows sign language, without its also being the case that some monkey knows sign language, but this would be a mistake. It is possible for the sentence $\forall x(Mx \rightarrow Sx)$ to be true even though the sentence $\exists x(Mx \,\&\, Sx)$ is false.

How can this be? The answer comes from considering whether these sentences would be true or false *if there were no monkeys*. If there were no monkeys at all (in the domain), then $\forall x(Mx \rightarrow Sx)$ would be *vacuously* true: take any monkey you like—it knows sign language! But if there were no monkeys at all (in the domain), then $\exists x(Mx \,\&\, Sx)$ would be false.

Another example will help to bring this home. Suppose we extend the above symbolization key, by adding:

Rx: $\_\_\_\_\_\__x$ is a refrigerator

Now consider the sentence $\forall x(Rx \rightarrow Mx)$. This symbolizes 'every refrigerator is a monkey'. This sentence is true, given our symbolization key, which is counterintuitive, since we (presumably) do not want to say that there are a whole bunch of refrigerator monkeys. It is important to remember, though, that $\forall x(Rx \rightarrow Mx)$ is true iff any member of the domain that is a refrigerator is a monkey. Since the domain is *animals*, there are no refrigerators in the domain. Again, then, the sentence is *vacuously* true.

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

When F is an empty predicate, a sentence $\forall x(Fx \rightarrow \ldots)$ will be vacuously true.

25.3 Picking a domain

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

8. Every rose has a thorn.

We might offer this symbolization key:

Rx: $\underline{\quad\quad}_x$ is a rose
Tx: $\underline{\quad\quad}_x$ has a thorn

It is tempting to say that sentence 8 should be symbolized as $\forall x(Rx \rightarrow Tx)$, but we have not yet chosen a domain. If the domain contains all roses, this would be a good symbolization. Yet if the domain is merely *things on my kitchen table*, then $\forall x(Rx \rightarrow Tx)$ would only come close to covering the fact that every rose *on my kitchen table* has a thorn. If there are no roses on my kitchen table, the sentence would be trivially true. This is not what we want. To symbolize sentence 8 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 8 can, if we like, be symbolized with $\forall x Tx$. This is true iff everything in the domain has a thorn; since the domain is just the roses, this is true iff every rose has a thorn. By restricting the domain, we have been able to symbolize our English sentence with a very short sentence of FOL. So this approach can save us trouble, if every sentence that we want to deal with is about roses.

Second, we can let the domain contain things besides roses: rhododendrons; rats; rifles; whatevers., and we will certainly need to include a more expansive domain if we simultaneously want to symbolize sentences like:

9. Every cowboy sings a sad, sad song.

Our domain must now include both all the roses (so that we can symbolize sentence 8) and all the cowboys (so that we can symbolize sentence 9). So we might offer the following symbolization key:

domain: people and plants

Cx: _____$_x$ is a cowboy

Sx: _____$_x$ sings a sad, sad song

Rx: _____$_x$ is a rose

Tx: _____$_x$ has a thorn

Now we will have to symbolize sentence 8 with $\forall x(Rx \rightarrow Tx)$, since $\forall x Tx$ would symbolize the sentence 'every person or plant has a thorn'. Similarly, we will have to symbolize sentence 9 with $\forall x(Cx \rightarrow Sx)$.

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

25.4 Ambiguous predicates

Suppose we just want to symbolize this sentence:

10. Adina is a skilled surgeon.

Let the domain be people, let Kx mean 'x is a skilled surgeon', and let a mean Adina. Sentence 10 is simply Ka.

Suppose instead that we want to symbolize this argument:

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

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

domain: people

Gx: _____$_x$ is greedy.

Hx: The hospital will hire _____$_x$.

Rx: _____$_x$ is a surgeon.

Kx: _____$_x$ is skilled.

b: Billy

Now the argument can be symbolized in this way:

- $\forall x\big[\neg(Rx \,\&\, Kx) \rightarrow \neg Hx\big]$
- $\forall x(Rx \rightarrow Gx)$
- $Rb \,\&\, \neg Kb$

- Therefore: $Gb \,\&\, \neg Hb$

Next suppose that we want to symbolize this argument:

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

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

- $(Rc \,\&\, Kc) \,\&\, Tc$
- Therefore: $Tc \,\&\, Kc$

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

- $(Rc \,\&\, K_1c) \,\&\, Tc$
- Therefore: $Tc \,\&\, K_2c$

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

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

Is it enough to have a predicate that means 'x is a skilled surgeon', rather than two predicates 'x is skilled' and 'x is a surgeon'? Sometimes. As sentence 10 shows, sometimes we do not need to distinguish between skilled surgeons and other surgeons.

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

CHAPTER 26

Multiple quantifiers

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

Consider

1. Someone loves everyone.

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

2. John loves everyone.

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

domain: all people
j: John
Lxy: $\_\_\_\__x$ loves $\_\_\_\__y$

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

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

26.1 The order of quantifiers

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

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

Sentence 3 can be symbolized by $\forall x \exists y Lxy$, and would be true of a love-triangle. For example, suppose that our domain of discourse is restricted to Imre, Juan and Karl. Suppose also that Karl loves Imre but not Juan, that Imre loves Juan but not Karl, and that Juan loves Karl but not Imre. Then sentence 3 is true.

Sentence 4 is symbolized by $\exists y \forall x Lxy$. Sentence 4 is *not* true in the situation just described. Again, suppose that our domain of discourse is restricted to Imre, Juan and Karl. This requires that all of Juan, Imre and Karl converge on (at least) one object of love.

The point of the example is to illustrate that the order of the quantifiers matters a great deal. Indeed, to switch them around is called a *quantifier shift fallacy*. Here is an example, which comes up in various forms throughout the philosophical literature:

- For every person, there is some truth they cannot know. ($\forall\exists$)
- Therefore: There is some truth that no person can know. ($\exists\forall$)

This argument form is obviously invalid. It's just as bad as:[1]

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

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

26.2 Stepping-stones to symbolization

Once we have the possibility of multiple quantifiers, representation in FOL can quickly start to become a bit tricky. When you are trying to symbolize a complex sentence, we recommend laying down several stepping stones. As usual, this idea is best illustrated by example. Consider this representation key:

domain: people and dogs

[1] Thanks to Rob Trueman for the example.

Dx: $\_\_\_\_\__{x}$ is a dog
Fxy: $\_\_\_\_\__{x}$ is a friend of $\_\_\_\_\__{y}$
Oxy: $\_\_\_\_\__{x}$ owns $\_\_\_\_\__{y}$
g: Geraldo

Now let's try to symbolize these sentences:

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

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

Sentence 6 can be paraphrased as, 'There is some y such that y is a dog owner'. Dealing with part of this, we might write $\exists y(y$ is a dog owner$)$. Now the fragment we have left as 'y is a dog owner' is much like sentence 5, except that it is not specifically about Geraldo. So we can symbolize sentence 6 by:

$$\exists y \exists x(Dx \,\&\, Oyx)$$

We should pause to clarify something here. In working out how to symbolize the last sentence, we wrote down $\exists y(y$ is a dog owner$)$. To be very clear: this is *neither* an FOL sentence *nor* an English sentence: it uses bits of FOL ($\exists$, y) and bits of English ('dog owner'). It is really is *just a stepping-stone* on the way to symbolizing the entire English sentence with a FOL sentence. You should regard it as a bit of rough-working-out, on a par with the doodles that you might absent-mindedly draw in the margin of this book, whilst you are concentrating fiercely on some problem.

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

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

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

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

for we would here have a *clash of variables*. The scope of the universal quantifier, $\forall x$, is the entire conditional, so the x in Dx should be governed by that, but Dx also falls under the scope of the existential quantifier $\exists x$, so the x in Dx should be governed by that. Now confusion reigns: which x are we talking about? Suddenly the sentence becomes ambiguous (if it is even meaningful at all), and logicians hate ambiguity. The broad moral is that a single variable cannot serve two quantifier-masters simultaneously.

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

$$\forall x\big[Fxg \rightarrow \exists z(Dz \mathbin{\&} Oxz)\big]$$

This adequately symbolizes sentence 7.

Sentence 8 can be paraphrased as 'For any x that is a dog owner, there is a dog owner who x is a friend of'. Using our stepping-stone tactic, this becomes

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

Completing the symbolization, we end up with

$$\forall x\big[\exists z(Dz \mathbin{\&} Oxz) \rightarrow \exists y\big(\exists z(Dz \mathbin{\&} Oyz) \mathbin{\&} Fxy\big)\big]$$

Note that we have used the same letter, z, in both the antecedent and the consequent of the conditional, but that these are governed by two different quantifiers. This is ok: there is no clash here, because it is clear which quantifier that variable falls under. We might graphically represent the scope of the quantifiers thus:

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

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

Sentence 9 is the trickiest yet. First we paraphrase it as 'For any x that is a friend of a dog owner, x owns a dog which is also owned by a friend of x'. Using our stepping-stone tactic, this becomes:

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

Breaking this down a bit more:

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

And a bit more:

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

And we are done!

26.3 Supressed quantifiers

Logic can often help to get clear on the meanings of English claims, especially where the quantifiers are left implicit or their order is ambiguous or unclear. The clarity of expression and thinking afforded by FOL can give you a significant advantage in argument, as can be seen in the following takedown by British political philosopher Mary Astell (1666–1731) of her contemporary, the theologian William Nicholls. In Discourse IV: The Duty of Wives to their Husbands of his *The Duty of Inferiors towards their Superiors, in Five Practical Discourses* (London 1701), Nicholls argued that women are naturally inferior to men. In the preface to the 3rd edition of her treatise *Some Reflections upon Marriage, Occasion'd by the Duke and Duchess of Mazarine's Case; which is also considered,* Astell responded as follows:

> 'Tis true, thro' Want of Learning, and of that Superior Genius which Men as Men lay claim to, she [Astell] was ignorant of the *Natural Inferiority* of our Sex, which our Masters lay down as a Self-Evident and Fundamental Truth. She saw nothing in the Reason of Things, to make this either a Principle or a Conclusion, but much to the contrary; it being Sedition at least, if not Treason to assert it in this Reign.
>
> For if by the Natural Superiority of their Sex, they mean that *every* Man is by Nature superior to *every* Woman, which is the obvious meaning, and that which must be stuck to if they would speak Sense, it wou'd be a Sin in *any* Woman to have Dominion over *any* Man, and the greatest Queen ought not to command but to obey her Footman, because

> no Municipal Laws can supersede or change the Law of Nature; so that if the Dominion of the Men be such, the *Salique Law*,[2] as unjust as *English Men* have ever thought it, ought to take place over all the Earth, and the most glorious Reigns in the *English, Danish, Castilian*, and other Annals, were wicked Violations of the Law of Nature!
>
> If they mean that *some* Men are superior to *some* Women this is no great Discovery; had they turn'd the Tables they might have seen that *some* Women are Superior to *some* Men. Or had they been pleased to remember their Oaths of Allegiance and Supremacy, they might have known that *One* Woman is superior to *All* the Men in these Nations, or else they have sworn to very little purpose.[3] And it must not be suppos'd, that their Reason and Religion wou'd suffer them to take Oaths, contrary to the Laws of Nature and Reason of things.[4]

We can symbolize the different interpretations Astell offers of Nicholls' claim that men are superior to women: He either meant that every man is superior to every woman, i.e.,

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

or that some men are superior to some women,

$$\exists x(Mx \mathbin{\&} \exists y(Wy \mathbin{\&} Sxy)).$$

The latter is true, but so is

$$\exists y(Wy \mathbin{\&} \exists x(Mx \mathbin{\&} Syx)).$$

(some women are superior to some men), so that would be "no great discovery." In fact, since the Queen is superior to all her subjects, it's even true that some woman is superior to every man, i.e.,

$$\exists y(Wy \wedge \forall x(Mx \rightarrow Syx)).$$

But this is incompatible with the "obvious meaning" of Nicholls' claim, i.e., the first reading. So what Nicholls claims amounts to treason against the Queen!

---

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

[3] In 1706, England was ruled by Queen Anne.

[4] Mary Astell, *Reflections upon Marriage*, 1706 Preface, iii–iv, and Mary Astell, *Political Writings*, ed. Patricia Springborg, Cambridge University Press, 1996, 9–10.

CHAPTER 27

Sentences of FOL

We will now carefully introduce what it is to be a sentence of FOL.

27.1 Vocabulary of FOL

We'll start by summarising, a bit more formally, the vocabulary of FOL. What can sentences of FOL be built from.

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

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

Variables x, y, z, or with subscripts, as needed $x_1, y_1, z_1, x_2, \ldots$. u, v, w may also be used.

---

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

Zero-Place Predicates = Atomic sentences of TFL $A^0, B^0, \ldots, Z^0$, with subscripts, as needed: $A^0_1, Z^0_2, A^0_{25}, J^0_{375}, \ldots$

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

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

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

etc. We drop the superscripts for ease.

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

Brackets (,)

Quantifiers $\forall, \exists$

27.2 Formulas

In §5, we went straight from the statement of the vocabulary of TFL to the definition of a sentence of TFL. In FOL, we will 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.

As we did for TFL, we will present a recursive definition of a formula of FOL.

The starting point of this is the notion of an *atomic formula*. In TFL we stared our definition with the notion of an atomic sentence, which were just given to us in our vocabulary. In FOL, the starting point of our definition is the notion of an atomic formula. Atomic formulas will be given by the following definition:

If $\mathcal{P}$ is an n-place predicate and $t_1, \ldots t_n$ are either variables or names, then $\mathcal{P}t_1 \ldots t_n$ is an ATOMIC FORMULA.

For example, if D is a one-place predicate (we might have introduced it to symbolise '_____$_x$ is a dog'), and L is a two-place predicate (we might have introduced it to symbolise '_____$_x$ loves _____$_y$'), then the following are atomic formulas:

$$Db, Dx, Dy, Lki, Lkx, Lyz$$

Formulas are constructed by starting with these and using either our TFL connectives or our quantifiers.

We can now give the recursive definition of what it is to be a formula of FOL.

1. If P is an n-place predicate and $t_1, \ldots t_n$ are either variables or names, then $Pt_1 \ldots t_n$ is a formula.
   These are called ATOMIC FORMULAS.

2. If X is a formula, then $\neg X$ is a formula.

3. If X and Y are formulas, then

   a) $(X \,\&\, Y)$ is a formula,
   b) $(X \vee Y)$ is a formula,
   c) $(X \rightarrow Y)$ is a formula, and
   d) $(X \leftrightarrow Y)$ is a formula.

4. If X is a formula, v is a variable and $\forall v$ and $\exists v$ do not already appear in X, then

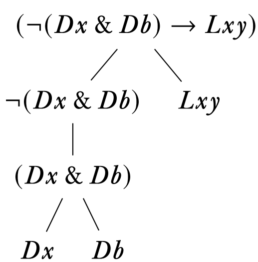
   a) $\forall v X$ is a formula
   b) $\exists v X$ is a formula.

5. Nothing else is a formula.

As for TFL, we start out with some formulas, such as Dx or Db, and we can construct more complicated formulas with our connectives, e.g.

$$(Dx \,\&\, Db),$$
$$\neg(Dx \,\&\, Db)$$
$$(\neg(Dx \,\&\, Db) \rightarrow Lxy)$$

And we can display their construction using our formation trees, as in 5.1.



This is exactly as in the case of TFL, the only difference is that the "leaves" of the tree have more structure to them: they're predicates applied to names or variables rather than simply the single atomic sentences that we had in TFL.

The new clauses here are in 4. This lets us put $\forall x$ in front of a formula, e.g. Bx to construct a formula $\forall x Bx$. We can also add quantifiers when the formula was already more complicated, e.g., we can construct a formula

$$\forall x(\neg(Dx \,\&\, Db) \to Lxy).$$

We could also have added an existential quantifier, $\exists$ to construct

$$\exists x(\neg(Dx \,\&\, Db) \to Lxy).$$

We can also do it with other variables, e.g.

$$\forall y(\neg(Dx \,\&\, Db) \to Lxy).$$

We can then add further quantifiers to *these* new formula, to construct, e.g.

$$\exists y \forall x(\neg(Dx \,\&\, Db) \to Lxy).$$

We can again display the structure and construction of the sentence perspicuously by presenting a formation tree:

$$\begin{array}{c}
\exists y \forall x(\neg(Dx \,\&\, Db) \to Lxy) \\
| \\
\forall x(\neg(Dx \,\&\, Db) \to Lxy) \\
| \\
(\neg(Dx \,\&\, Db) \to Lxy) \\
\begin{array}{cc} / & \backslash \\ \begin{array}{c} \neg(Dx \,\&\, Db) \\ | \\ (Dx \,\&\, Db) \\ \begin{array}{cc} / & \backslash \\ Dx & Db \end{array} \end{array} & Lxy \end{array}
\end{array}$$

One more example:

$$\begin{tikzcd}[arrows=dash]
& \exists z \forall y (Ryz \mathbin{\&} \exists x Qx) \arrow[d] & \\
& \forall y (Ryz \mathbin{\&} \exists x Qx) \arrow[d] & \\
& (Ryz \mathbin{\&} \exists x Qx) \arrow[dl] \arrow[dr] & \\
Ryz & & \exists x Qx \arrow[d] \\
& & Qx
\end{tikzcd}$$

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

Why in 4 did we have the restriction that $\exists v$ or $\forall v$ is not already in X? This is to ensure that variables only serve one master at any one time (see §26). Otherwise we could see that $\forall x Rxx$ is a sentence and then conclude that $\exists x \forall x Rxx$, which is not a good sentence. However, note that $\exists x Cx \lor \forall x Bx$ is a sentence. Here the variable x is used in both quantifiers, but there's no ambiguity because the sentence was constructed by combing the two sentences $\exists x Cx$ and $\forall x Bx$ with a connective, $\lor$. The *scope* of $\exists x$ is just $\exists x Cx$; it doesn't look "over" the connective to where $\forall x$ is used. So the two uses of x are kept separate and no problems arise. However, to avoid any potential worries it is generally a good idea to use different variables when symbolising sentences; in this case one could equally well give $\exists x Cx \lor \forall y By$ as a symbolisation.

The notions of scope and main logical operators that were given in 5 equally applies to FOL but now the main logical operator might be a quantifier. These were:

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

We can graphically illustrate scopes as follows:

$$\overbrace{\exists z \underbrace{\forall y(\overbrace{\exists x Qx}^{\text{scope of } \exists x} \,\&\, Ryz)}_{\text{scope of } \forall y}}^{\text{scope of } \exists z}$$

27.3 Sentences

Recall that we are largely concerned in logic with assertoric sentences: sentences that can be either true or false. Many formulas are not sentences. Consider the following symbolization key:

domain: people
Lxy: $\_\_\_\_\__x$ loves $\_\_\_\_\__y$
b: Boris

Consider the atomic formula Lzz. Can it be true or false? You might think that it will be true just in case the person named by z loves themself, in the same way that Lbb is true just in case Boris (the person named by b) loves himself. *However, z is a variable, and does not name anyone or any thing.*

Of course, if we put an existential quantifier out front, obtaining $\exists zLzz$, then this would be true iff someone loves herself. Equally, if we wrote $\forall zLzz$, this would be true iff everyone loves themself. The point is that we need a quantifier to tell us how to deal with a variable.

Let's make this idea precise.

> A BOUND VARIABLE is an occurrence of a variable v that is within the scope of either $\forall v$ or $\exists v$.
>
> A FREE VARIABLE is any variable that is not bound.

For example, consider the formula

$$\forall x(Ex \lor Dy) \to \exists z(Ex \to Lzx)$$

The scope of the universal quantifier $\forall x$ is $\forall x(Ex \lor Dy)$, so the first x is bound by the universal quantifier. However, the second and third occurrence of x are free. Equally, the y is free. The scope of the existential quantifier $\exists z$ is $(Ex \to Lzx)$, so z is bound.

Finally we can say the following.

A SENTENCE of FOL is any formula of FOL that contains no free variables.

27.4 Bracketing conventions

We will adopt the same notational conventions governing brackets that we did for TFL (see §5 and §11.3.): we may omit the outermost brackets of a formula.

CHAPTER 28

Ambiguity

In chapter 7 we discussed the fact that sentences of English can be ambiguous, and pointed out that sentences of TFL are not. One important application of this fact is that the structural ambiguity of English sentences can often, and usefully, be straightened out using different symbolizations. One common source of ambiguity is *scope ambiguity*, where the English sentence does not make it clear which logical word is supposed to be in the scope of which other. Multiple interpretations are possible. In FOL, every connective and quantifier has a well-determined scope, and so whether or not one of them occurs in the scope of another in a given sentence of FOL is always determined.

For instance, consider the English idiom,

1. Everything that glitters is not gold.

If we think of this sentence as of the form 'every F is not G' where Fx symbolizes '$\_\_\_\_\__x$ glitters' and Gx is '$\_\_\_\_\__x$ is not gold', we would symbolize it as:

- $\forall x(Fx \rightarrow \neg Gx)$,

in other words, we symbolize it the same way as we would 'Nothing that glitters is gold'. But the idiom does not mean that! It means that one should not assume that just because something glitters, it is gold; not everything that appears valuable is in fact valuable. To capture the actual meaning of the idiom, we would have to symbolize it instead as we would 'Not everything that glitters is gold', i.e., in the following way:

- $\neg\forall x(Fx \rightarrow Gx)$

Compare the first of these with the previous symbolization: again we see that the difference in the two meanings of the ambiguous sentence lies in whether the '$\neg$' is in the scope of the '$\forall$' (in the first symbolization) or '$\forall$' is in the scope of '$\neg$' (in the second).

Of course we can alternatively symbolize the two readings using existential quantifiers as well:

- $\neg\exists x(Fx \,\&\, Gx)$
- $\exists x(Fx \,\&\, \neg Gx)$

In chapter ?? we discussed how to symbolize sentences involving 'only'. Consider the sentence:

2. Only young cats are playful.

According to our schema, we would symbolize it this way:

- $\forall x(Px \rightarrow (Yx \,\&\, Cx))$

The meaning of this sentence of FOL is something like, 'If an animal is playful, it is a young cat'. (Assuming that the domain is animals, of course.) This is probably not what's intended in uttering sentence 2, however. It's more likely that we want to say that old cats are not playful. In other words, what we mean to say is that if something is a cat and playful, it must be young. This would be symbolized as:

- $\forall x((Cx \,\&\, Px) \rightarrow Yx)$

There is even a third reading! Suppose we're talking about young animals and their characteristics. And suppose you wanted to say that of all the young animals, only the cats are playful. You could symbolize this reading as:

- $\forall x((Yx \,\&\, Px) \rightarrow Cx)$

Each of the last two readings can be made salient in English by placing the stress appropriately. For instance, to suggest the last reading, you would say 'Only young *cats* are playful', and to get the other reading you would say 'Only *young* cats are playful'. The very first reading can be indicate by stressing both 'young' and 'cats': 'Only *young cats* are playful' (but not old cats, or dogs of any age).

In sections ?? and ?? we discussed the importance of the order of quantifiers. This is relevant here because, in English, the order of quantifiers is sometimes not completely determined. When both universal (‘all’) and existential (‘some’, ‘a’) quantifiers are involved, this can result in scope ambiguities. Consider:

3. Everyone went to see a movie.

This sentence is ambiguous. In one interpretatation, it means that there is a single movie that everyone went to see. In the other, it means that everyone went to see some movie or other, but not necessarily the same one. The two readings can be symbolized, respectively, by

- $\exists x(Mx \,\&\, \forall y(Py \rightarrow Sy,x))$
- $\forall y(Py \rightarrow \exists x(Mx \,\&\, Sy,x))$

We assume here that the domain contains (at least) people and movies, and the symbolization key,

Py: $\underline{\qquad}_y$ is a person,
Mx: $\underline{\qquad}_x$ is a movie
Sy,x: $\underline{\qquad}_y$ went to see $\underline{\qquad}_x$.

In the first reading, we say that the existential quantifier has *wide scope* (and its scope contains the universal quantifier, which has *narrow scope*), and the other way round in the second.

In chapter 39, we encountered another scope ambiguity, arising from definite descriptions interacting with negation. Consider Russell’s own example:

4. The King of France is not bald.

If the definite description has wide scope, and we are interpreting the ‘not’ as an ‘inner’ negation (as we said before), sentence 4 is interpreted to assert the existence of a single King of France, to whom we are ascribing non-baldness. In this reading, it is symbolized as ‘$\exists x\big[Kx \,\&\, \forall y(Ky \rightarrow x = y)) \,\&\, \neg Bx\big]$’. In the other reading, the ‘not’ denies the sentence ‘The King of France is bald’, and we would symbolize it as: ‘$\neg\exists x\big[Kx \,\&\, \forall y(Ky \rightarrow x = y)) \,\&\, Bx\big]$’. In the first case, we say that the definite description has wide scope and in the second that it has narrow scope.

PART VI

Interpretations

CHAPTER 29

Extensionality

Recall that TFL is a truth-functional language. Its connectives are all truth-functional, and *all* that we can do with TFL is key sentences to particular truth values. We can do this *directly*. For example, we might stipulate that the TFL sentence P is to be true. Alternatively, we can do this *indirectly*, offering a symbolization key, e.g.:

P: Big Ben is in London

Now recall from §10 that this should be taken to mean:

- The TFL sentence P is to take the same truth value as the English sentence 'Big Ben is in London' (whatever that truth value may be)

The point that we emphasized is that TFL cannot handle differences in meaning that go beyond mere differences in truth value.

29.1 Symbolizing versus translating

FOL has some similar limitations, but it goes beyond mere truth values, since it enables us to split up sentences into terms, predicates and quantifier expressions. This enables us to consider what is *true of* some particular object, or of some or all objects. But we can do no more than that.

When we provide a symbolization key for some FOL predicates, such as:

Cx: $\underline{\hspace{3em}}_x$ teaches Logic III in Calgary we do not carry the *meaning* of the English predicate across into our FOL predicate. We are simply stipulating something like the following:

- Cx and '$\_\_\_\_\__x$ teaches Logic III in Calgary' are to be *true of* exactly the same things.

So, in particular:

- Cx is to be true of all and only those things which teach Logic III in Calgary (whatever those things might be).

This is an indirect stipulation. Alternatively, we can directly stipulate which objects a predicate should be true of. For example, we can stipulate that Cx is to be true of Richard Zach, and Richard Zach alone. As it happens, this direct stipulation would have the same effect as the indirect stipulation. Note, however, that the English predicates '_____ is Richard Zach' and '_____ teaches Logic III in Calgary' have very different meanings!

The point is that FOL does not give us any resources for dealing with nuances of meaning. When we interpret FOL, all we are considering is what the predicates are true of, regardless of whether we specify these things directly or indirectly. The things a predicate is true of are known as the EXTENSION of that predicate. We say that FOL is an EXTENSIONAL LANGUAGE because FOL does not represent differences of meaning between predicates that have the same extension.

For this reason, we say only that FOL sentences *symbolize* English sentences. It is doubtful that we are *translating* English into FOL, as translations should preserve meanings, and not just extensions.

29.2 A word on extensions

We can stipulate directly what predicates are to be true of, so it is worth noting that our stipulations can be as arbitrary as we like. For example, we could stipulate that Hx should be true of, and only of, the following objects:

Justin Trudeau
the number π
every top-F key on every piano ever made

Now, the objects that we have listed have nothing particularly in common. But this doesn't matter. Logic doesn't care about what strikes us mere humans as ‘natural’ or ‘similar’. Armed with this interpretation of Hx, suppose we now add to our symbolization key:

j: Justin Trudeau
r: Rachel Notley
p: the number π

Then Hj and Hp will both be true, on this interpretation, but Hr will be false, since Rachel Notley was not among the stipulated objects.

29.3 Many-place predicates

All of this is quite easy to understand when it comes to one-place predicates, but it gets messier when we consider two-place predicates. Consider a symbolization key like:

Lxy: $\underline{\quad\quad}_x$ loves $\underline{\quad\quad}_y$

Given what we said above, this symbolization key should be read as saying:

- Lxy and ‘$\underline{\quad\quad}_x$ loves $\underline{\quad\quad}_y$’ are to be true of exactly the same things

So, in particular:

- Lxy is to be true of x and y (in that order) iff x loves y.

It is important that we insist upon the order here, since love—famously—is not always reciprocated. (Note that ‘x’ and ‘y’ here are symbols of augmented English, and that they are being *used*. By contrast, x and y are symbols of FOL, and they are being *mentioned*.)

That is an indirect stipulation. What about a direct stipulation? This is slightly harder. If we *simply* list objects that fall under Lxy, we will not know whether they are the lover or the beloved (or both). We have to find a way to include the order in our explicit stipulation.

To do this, we can specify that two-place predicates are true of *pairs* of objects, where the order of the pair is important. Thus we might stipulate that Bxy is to be true of, and only of, the following pairs of objects:

⟨Lenin, Marx⟩
⟨Heidegger, Sartre⟩
⟨Sartre, Heidegger⟩

Here the angle-brackets keep us informed concerning order. Suppose we now add the following stipulations:

l: Lenin
m: Marx
h: Heidegger
r: Sartre

Then Blm will be true, since ⟨Lenin, Marx⟩ was in our explicit list, but Bml will be false, since ⟨Marx, Lenin⟩ was not in our list. However, both Bhr and Brh will be true, since both ⟨Heidegger, Sartre⟩ and ⟨Sartre, Heidegger⟩ are in our explicit list.

To make these ideas more precise, we would need to develop some *set theory*. That would give us some precise tools for dealing with extensions and with ordered pairs (and ordered triples, etc.). However, set theory is not covered in this book, so we will leave these ideas at an imprecise level. Nevertheless, the general idea should be clear.

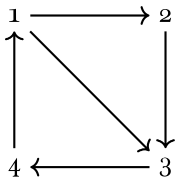
29.4 Interpretation

We defined a VALUATION in TFL as any assignment of truth and falsity to atomic sentences. In FOL, we are going to define an INTERPRETATION as consisting of three things:

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

The symbolization keys that we considered in Part V consequently give us one very convenient way to present an interpretation. We will continue to use them throughout this chapter. However, it is sometimes also convenient to present an interpretation *diagrammatically*.

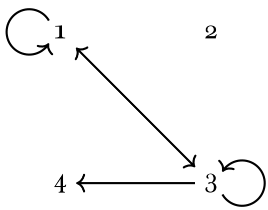
Suppose we want to consider just a single two-place predicate, Rxy. Then we can represent it just by drawing an arrow between two objects, and stipulate that Rxy is to hold of x and y just in case there is an arrow running from x to y in our diagram. As an example, we might offer:



This would be suitable to characterize an interpretation whose domain is the first four positive whole numbers, and which interprets Rxy as being true of and only of:

$$\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle, \langle 4, 1\rangle, \langle 1, 3\rangle$$

Equally we might offer:



for an interpretation with the same domain, which interprets Rxy as being true of and only of:

$$\langle 1, 3\rangle, \langle 3, 1\rangle, \langle 3, 4\rangle, \langle 1, 1\rangle, \langle 3, 3\rangle$$

If we wanted, we could make our diagrams more complex. For example, we could add names as labels for particular objects. Equally, to symbolize the extension of a one-place predicate, we might simply draw a ring around some particular objects and stipulate that the thus encircled objects (and only them) are to fall under the predicate Hx, say.

CHAPTER 30

Truth in FOL

We know what interpretations are. Since, among other things, they tell us which predicates are true of which objects, they will provide us with an account of the truth of atomic sentences. However, we must also present a detailed account of what it is for an arbitrary FOL sentence to be true or false in an interpretation.

But we defined what a sentence was by first specifying what a formula is. Formulas like Hx aren't the sorts of things that are true or false in interpretations. Only sentences are true or false. But if we provide extra information we can determine the truth of Hx: we need to specify what x refers to. This is done by using a variable assignment:

> A variable assignment specifies an object for each variable.

We define whether a formula is true or false *under a variable assignment*. We know from §27 that there are three kinds of formulas in FOL:

- atomic formulas
- formulas whose main logical operator is a sentential connective
- formulas whose main logical operator is a quantifier

We need to explain truth for all three kinds of formula.

We will provide a completely general explanation in this section. However, to try to keep the explanation comprehensible, we will, at several points, use the following interpretation:

domain: all people born before 2000CE
a: Aristotle
b: Beyoncé

Px: $\_\_\_\_\__x$ is a philosopher
Rxy: $\_\_\_\_\__x$ was born before $\_\_\_\_\__y$

This will be our *go-to example* in what follows.

30.1 Atomic formulas

Atomic formulas are things like Px, Pb or Rax.

An atomic sentence like Pb is checked for truth just by consulting our interpretation: Px is '$\_\_\_\_\__x$ is a philosopher', so if we're looking at Pb we fill out the gap with Beyoncé, and Beyoncé is not a philosopher, so Pb is false.

What about Px? This reads something like 'they are a philosopher'. The question is who 'they' refers to, or in the logic terms: who x is. This depends on a variable assignment. Our variable assignment needs to give an object in our domain for the variable x. For example, it might give Beyoncé, then since Beyoncé is not a philosopher, Px would be false on this variable assignment. Our variable assignment doesn't need to specify one of the objects that are named, it can give us anyone in our domain, e.g. Queen Elizabeth II. Under the variable assignment which assigns x Queen Elizabeth II, Px is false: Queen Elizabeth II is not a philosopher.

Likewise, on this interpretation, Rab is true iff the object named by a was born before the object named by b. Well, Aristotle was born before Beyoncé. So Rab is true. Equally, Raa is false: Aristotle was not born before Aristotle. How about Rax? Well what does our variable assignment specify for x? If we have a variable assignment where x is Queen Elizabeth II, then Rax is true: Aristotle was born before Queen Elizabeth II.

Dealing with atomic sentences, then, is very intuitive. When R is an n-place predicate and t_1, $t_2 \ldots t_n$ are names or variables, then

> $Rt_1t_2 \ldots t_n$ is true in an interpretation under a variable assignment **iff**
> R is true of the objects referred to by $t_1, t_2, \ldots, t_n$ in that interpretation under that variable assignment (considered in that order)

30.2 Sentential connectives

We saw in §27 that FOL formulas can be built up from simpler ones using the truth-functional connectives that were familiar from TFL. The rules governing these truth-functional connectives are *exactly* the same as they were when we considered TFL. Here they are:

$X \,\&\, Y$ is true in an interpretation under a variable assignment
iff
both X and Y is true in that interpretation under that variable assignment

$X \vee Y$ is true in an interpretation under a variable assignment
iff
either X is true or Y is true in that interpretation under that variable assignment

$\neg X$ is true in an interpretation under a variable assignment **iff**
X is false in that interpretation under that variable assignment

$X \rightarrow Y$ is true in an interpretation under a variable assignment
iff
either X is false or Y is true in that interpretation under that variable assignment

$X \leftrightarrow Y$ is true in an interpretation under a variable assignment
iff
X has the same truth value as Y in that interpretation under that variable assignment

This is just another presentation of the truth rules we gave for the connectives in TFL; it just does so in a slightly different way. Some examples will probably help to illustrate the idea. On our go-to interpretation:

- Pa is true
- $Rab \,\&\, Pb$ is false because, although Rab is true, Pb is false
- $\neg Pa$ is false
- $Pa \,\&\, \neg(Pb \,\&\, Rab)$ is true, because Pa is true and Pb is false, so $Pb \,\&\, Rab$ is false, thus $\neg(Pb \,\&\, Rab)$ is also true.

Make sure you understand these examples.

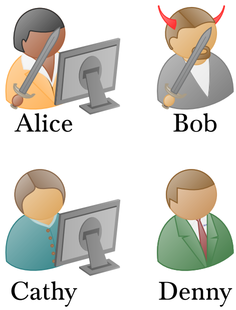
We can also carry variable assignments around with us. Consider a variable assignment which assigns David Hume to x. Then

- Px is true under this variable assignment: David Hume was a philosopher
- Bxa is false under this variable assignment: David Hume was born after Aristotle
- $Px \rightarrow Bxa$ is false under this variable assignment: Px is true and Bxa is false, so by our rule for $\rightarrow$, $Px \rightarrow Bxa$ is false.

30.3 When the main logical operator is a quantifier

The exciting innovation in FOL, though, is the use of *quantifiers*.

Consider the following interpretation:



domain: People in above picture (Alice, Bob, Cathy and Denny)
Hx: $\underline{\qquad}_x$ has horns (Bob)
Sx: $\underline{\qquad}_x$ carrying a sword (Alice and Bob)
Cx: $\underline{\qquad}_x$ is looking at a computer (Alice and Cathy)

Is $\exists x Sx$ true? To check this we see if there is a choice of an object for x which gives us a variable assignment under which Sx is true. Consider assigning Alice to x, which we can as shorthand write by $x \mapsto$ Alice. Under this variable assignment, Sx is true: Alice does have a sword. So $\exists x Sx$ is true. There is a choice of an object in our domain for x under which Sx is true.

What about $\forall x Sx$? This is true iff Sx is true under any choice of a person for x. Let's go through them.

	Sx
$x \mapsto$ Alice	T
$x \mapsto$ Bob	T
$x \mapsto$ Cathy	F
$x \mapsto$ Denny	F

So $\forall x Sx$ is false: it is not the case that Sx is true under any choice of an object for x: when we have an assignment of Cathy to x, Sx is false.

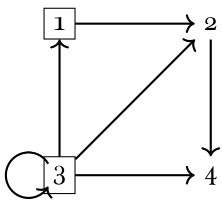
What about $\forall x(Sx \rightarrow (Hx \vee Cx))$

	Sx	Hx	Cx	$Hx \vee Cx$	$Sx \rightarrow (Hx \vee Cx)$
$x \mapsto$ Alice	T	F	T	T	T
$x \mapsto$ Bob	T	T	F	T	T
$x \mapsto$ Cathy	F	F	T	T	T
$x \mapsto$ Denny	F	F	F	F	T

So $Sx \rightarrow (Hx \vee Cx)$ is true under every assignment of an object to the variable x. And so $\forall x(Sx \rightarrow (Hx \vee Cx))$ is true.

We have to tread more carefully once we start having multiple quantifiers. Let's walk through some cases.

Consider a new interpretation:



domain: Numbers 1, 2, 3 and 4.
Rxy: There is an arrow from $\_\_\_\_\__x$ to $\_\_\_\_\__y$ in the diagram.
Sx: There is a square around $\_\_\_\_\__x$ in the diagram.

Is $\exists x \forall y Rxy$ true? We need to find some choice of an object for x where $\forall y Rxy$ is true under that choice ("variable assignment"). Let's (with foresight) chose the number 3 for x ($x \mapsto 3$). Is $\forall y Rxy$ true under this variable assignment $x \mapsto 3$? To check this we need to do something more with our variable assignment: we need to extend it with a choice of an object for y. Moreover, we need to think about all ways of picking an object for y, while we've fixed x as the number 3. Consider, e.g. $y \mapsto 1$. Then we have to evaluate whether Rxy is true with $x \mapsto 3, y \mapsto 1$. This is true iff there is an arrow from 3 (x) to 1 (y), and there is such an arrow. We can work through all the cases an see that all of them have an arrow from 3, so Rxy is true for any choice of object for y:

		Rxy
$x \mapsto 3$	$y \mapsto 1$	T
$x \mapsto 3$	$y \mapsto 2$	T
$x \mapsto 3$	$y \mapsto 3$	T
$x \mapsto 3$	$y \mapsto 4$	T

So under any way of extending our variable assignment of $x \mapsto 3$ by choosing an object for y results in a variable assignment on which Rxy true. This tells us that $\forall y Rxy$ is true under the variable assignment $x \mapsto 3$. And *that* tells us that $\exists x \forall y Rxy$ is true: there's an assignment of the variable x under which the constituent formula $\forall y Rxy$ is true.

What about $\exists x \exists y(Rxy \,\&\, Ryx)$? To show that it is true we will want to choose an object that we can assign to x under which $\exists y(Rxy \,\&\, Ryx)$ is true. Let's consider $x \mapsto 3$ (again I'm using my forsight of what will come to choose carefully). Now is $\exists y(Rxy \,\&\, Ryx)$ true under the variable assignment $x \mapsto 3$? We need to find an extension of this which chooses an object for y under which $Rxy \,\&\, Ryx$ is true. Consider $y \mapsto 3$. We now have a variable assignment $x \mapsto 3, y \mapsto 3$. They are different variables but there's nothing stopping them denoting the same object. And we can then consider whether $Rxy \,\&\, Ryx$ is true under this interpretation. Well, Rxy is true: 3 does have an arrow to 3. And Ryx is also true: 3 does have an arrow to 3. So by our clause for &, $Rxy \,\&\, Ryx$ is true under this variable assignment $x \mapsto 3, y \mapsto 3$. And so $\exists y(Rxy \,\&\, Ryx)$ is true under the variable assignment $x \mapsto 3$. And so $\exists x \exists y(Rxy \,\&\, Ryx)$ is true in this interpretation.

One more example: $\forall x(Sx \to \exists y Rxy)$? To check this is true we will need to go through each of our objects for x and see that $Sx \to \exists y Rxy$ is true under that interpretation. We can already see if Sx is true under each variable assignment, and if we find that Sx is false that's enough information to determine that $Sx \to \exists y Rxy$ is true (check the definition of truth for $\to$ to see this):

	Sx	$\exists y Rxy$	$Sx \to \exists y Rxy$
$x \mapsto 1$	T	?	?
$x \mapsto 2$	F	?	T
$x \mapsto 3$	T	?	?
$x \mapsto 4$	F	?	T

So we need to check whether $\exists y Rxy$ is true under the variable assignments $x \mapsto 1$ and $x \mapsto 3$.

Consider $x \mapsto 1$. We can find an object for y where Rxy is true under that variable assignment: consider $y \mapsto 2$. Since there is an arrow from 1 to 2, Rxy is true in the variable assignment $x \mapsto 1$ and $y \mapsto 2$. Thus $\exists y Rxy$ is true on the variable assignment $x \mapsto 1$. We're also able to do something similar for $x \mapsto 3$:

		Rxy
$x \mapsto 1$	$y \mapsto 2$	T
$x \mapsto 3$	$y \mapsto 3$	T

So we now have

	Sx	$\exists y Rxy$	$Sx \to \exists y Rxy$
$x \mapsto 1$	T	T	T
$x \mapsto 2$	F	?	T
$x \mapsto 3$	T	T	T
$x \mapsto 4$	F	?	T

So $\forall x(Sx \to \exists y Rxy)$ is true. Informally we might say this as: for every number that has a square around it has an arrow going out of it.

One final example: $\forall x \forall y Rxy$. To check this we will need to consider all choices for x and all choices for y and check Rxy is true on all of them. There are 16 such choices. But we won't have to go through them all: it'll be false. Consider $x \mapsto 1$ and $y \mapsto 4$. Rxy is false under this variable assignment: there is no arrow from 1 to 4. Thus $\forall y Rxy$ is false on the variable assignment $x \mapsto 1$. And so $\forall x \forall y Rxy$ is false in the interpretation.

Let's now give a formal definition of the idea we've been using here. Quantified formulas like $\exists y Rxy$ still need to be given truth conditions *relative to a variable assignment*, because we'll need to specify an assignment of an object for the free variable x. So we define when $\exists v X$ is true under a variable assignment α, which might be, e.g. $x \mapsto 1$. To give a general definition, though we might also consider whether $\forall y Rxy$ is true under a variable assignment $x \mapsto 1, y \mapsto 4$. This is slightly odd: we're considering $\exists y Rxy$ but have been told who y refers to already. However when we evaluate it we simply ignore whatever our given variable assignment tells us about y: we consider variable assignments that modify the assignment by changing what is assigned to y. And by modifying it to $x \mapsto 1, y \mapsto 2$ we have a variable assignment under which

Rxy is true, so $\exists y Rxy$ is true under the original variable assignment $x \mapsto 1, y \mapsto 4$.

Similarly consider $\exists x Sx$ under the variable assignment $x \mapsto 1$. To evaluate this we consider modification of this variable assignment which assign other objects to x. Under the variable assignment $x \mapsto 2$, Sx is true. So under our original variable assignment $\exists x Sx$ is true: it didn't matter what our original variable assignment was, we ignored this and considered variants to evaluate its truth.

This is a general feature: Sentences, which have all variables bound, have truth values independent of any variable assignment they're evaluated with: when we have a quantifier like $\exists x$ or $\forall x$ we ignore whatever our original variable assignment told us about x. So when all our variables are bound by quantifiers, all the original components of our variable assignment are ignored. To summarise: Sentences are simply true or false in interpretations, variable assignments don't matter.

Now for our formal definition:

$\forall v X$ is true under a variable assignment α
iff X is true under *every* variable assignment that is the result of modifying/extending α with a choice of an object in our domain for v.

$\exists v X$ is true under a variable assignment α
iff X is true under *some* variable assignment that is the result of modifying/extending α with a choice of an object in our domain for v.

To be clear: all this is doing is formalizing the intuitive idea expressed in our examples. The result is a bit ugly, and the final definition might look a bit opaque. Hopefully, though, the *spirit* of the idea is clear.

CHAPTER 31

Semantic concepts

Offering a precise definition of truth in FOL was more than a little fiddly, but now that we are done, we can define various central logical notions. These will look very similar to the definitions we offered for TFL. However, remember that they concern *interpretations*, rather than valuations.

$$X_1, X_2, \ldots, X_n \therefore Z$$

is VALID iff there is no interpretation in which all of $X_1, X_2, \ldots, X_n$ are true and in which Z is false.

The other logical notions also have corresponding definitions in FOL:

- An FOL sentence X is a LOGICAL TRUTH iff X is true in every interpretation.
- X is a CONTRADICTION iff X is false in every interpretation.
- Two FOL sentences X and Y are LOGICALLY EQUIVALENT iff they are true in exactly the same interpretations as each other.
- The FOL sentences $X_1, X_2, \ldots, X_n$ are JOINTLY LOGICALLY CONSISTENT iff there is some interpretation in which all of the sentences are true. They are JOINTLY LOGICALLY INCONSISTENT iff there is no such interpretation.

CHAPTER 32

Using interpretations

32.1 Logical truths and contradictions

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 will start with a domain that has just one member. For concreteness, let's say it is the city of Paris.

domain: Paris

The name d must refer to something in the domain, so we have no option but:

d: Paris

Recall that we want $\exists x Axx$ to be true, so we want all members of the domain to be paired with themselves in the extension of A. We can just offer:

Axy: $\_\_\_\_\__x$ is identical with $\_\_\_\_\__y$

Now Add is true, so it is surely true that $\exists x Axx$. Next, we want Bd to be false, so the referent of d must not be in the extension of B. We might simply offer:

Bx: $\_\_\_\_\__x$ is in Germany

Now we have an interpretation where $\exists x Axx$ is true, but where Bd is false. So there is an interpretation where $\exists x Axx \rightarrow Bd$ is false. So $\exists x Axx \rightarrow Bd$ is not a logical truth.

We can just as easily show that $\exists x Axx \rightarrow Bd$ is not a contradiction. We need only specify an interpretation in which $\exists x Axx \rightarrow Bd$ is true; i.e., an interpretation in which either $\exists x Axx$ is false or Bd is true. Here is one:

domain: Paris
d: Paris
Axy: $\_\_\_\_\__x$ is identical with $\_\_\_\_\__y$
Bx: $\_\_\_\_\__x$ is in France

This shows that there is an interpretation where $\exists x Axx \rightarrow Bd$ is true. So $\exists x Axx \rightarrow Bd$ is not a contradiction.

32.2 Logical equivalence

Suppose we want to show that $\forall x Sx$ and $\exists x Sx$ are not logically equivalent. We need to construct an interpretation in which the two sentences have different truth values; we want one of them to be true and the other to be false. We start by specifying a domain. Again, we make the domain small so that we can specify extensions easily. In this case, we will need at least two objects. (If we chose a domain with only one member, the two sentences would end up with the same truth value. In order to see why, try constructing some partial interpretations with one-member domains.) For concreteness, let's take:

domain: Ornette Coleman, Miles Davis

We can make $\exists x Sx$ true by including something in the extension of S, and we can make $\forall x Sx$ false by leaving something out of the extension of S. For concreteness we will offer:

Sx: $\_\_\_\_\__x$ plays saxophone

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

> To show that X is not a logical truth, it suffices to find an interpretation where X is false.
> To show that X is not a contradiction, it suffices to find an interpretation where X is true.
> To show that X and Y are not logically equivalent, it suffices to find an interpretation where one is true and the other is false.

32.3 Validity, entailment and consistency

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

Consider the following argument in FOL:

$$\exists x(Gx \rightarrow Ga) \therefore \exists x Gx \rightarrow Ga$$

To show that this is invalid, we must make the premise true and the conclusion false. The conclusion is a conditional, so to make it false, the antecedent must be true and the consequent must be false. Clearly, our domain must contain two objects. Let's try:

domain: Karl Marx, Ludwig von Mises
Gx: $\underline{\quad\quad}_x$ hated communism
a: Karl Marx

Given that Marx wrote *The Communist Manifesto*, Ga is plainly false in this interpretation. But von Mises famously hated communism, so $\exists x Gx$ is true in this interpretation. Hence $\exists x Gx \rightarrow Ga$ is false, as required.

Does this interpretation make the premise true? Yes it does! Note that $Ga \rightarrow Ga$ is true. (Indeed, it is a logical truth.) But then certainly

$\exists x(Gx \to Ga)$ is true, so the premise is true, and the conclusion is false, in this interpretation. The argument is therefore invalid.

In passing, note that we have also shown that $\exists x(Gx \to Ga)$ does *not* entail $\exists xGx \to Ga$. 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:

$$\forall x\exists yLxy \therefore \exists y\forall xLxy$$

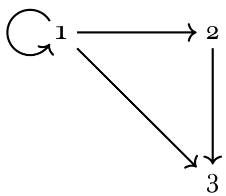
Again, we want to show that this is invalid. To do this, we must make the premises true and the conclusion false. Here is a suggestion:

domain: UK citizens currently in a civil partnership with another UK citizen

Lxy: $\underline{\quad\quad}_x$ is in a civil partnership with $\underline{\quad\quad}_y$

The premise is clearly true on this interpretation. Anyone in the domain is a UK citizen in a civil partnership with some other UK citizen. That other citizen will also, then, be in the domain. So for everyone in the domain, there will be someone (else) in the domain with whom they are in a civil partnership. Hence $\forall x\exists yLxy$ is true. However, the conclusion is clearly false, for that would require that there is some single person who is in a civil partnership with everyone in the domain, and there is no such person, so the argument is invalid. We observe immediately that the sentences $\forall x\exists yLxy$ and $\neg\exists y\forall xLxy$ are jointly consistent and that $\forall x\exists yLxy$ does not entail $\exists y\forall xLxy$.

For our third example, we'll mix things up a bit. In §29, we described how we can present some interpretations using diagrams. For example:



Using the conventions employed in §29, the domain of this interpretation is the first three positive whole numbers, and Rxy is true of x and y just in case there is an arrow from x to y in our diagram. Here are some sentences that the interpretation makes true:

- $\forall x\exists yRyx$

- $\exists x \forall y Rxy$ witness 1
- $\exists x \forall y (Ryx \leftrightarrow x = y)$ witness 1
- $\exists x \exists y \exists z ((\neg y = z \,\&\, Rxy) \,\&\, Rzx)$ witness 2
- $\exists x \forall y \neg Rxy$ witness 3
- $\exists x (\exists y Ryx \,\&\, \neg \exists y Rxy)$ 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.:

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

and many more besides.

> To show that $X_1, X_2, \ldots, X_n \therefore Z$ is invalid, it suffices to find an interpretation where all of $X_1, X_2, \ldots, X_n$ are true and where Z is false.
> That same interpretation will show that $X_1, X_2, \ldots, X_n$ do not entail Z.
> It will also show that $X_1, X_2, \ldots, X_n, \neg Z$ are jointly consistent.

When you provide an interpretation to refute a claim—to logical truth, say, or to entailment—this is sometimes called providing a *counter-interpretation* (or providing a *counter-model*).

CHAPTER 33

Reasoning about all interpretations

33.1 Logical truths and contradictions

We can show that a sentence is *not* a logical truth just by providing one carefully specified interpretation: an interpretation in which the sentence is false. To show that something is a logical truth, on the other hand, it would not be enough to construct ten, one hundred, or even a thousand interpretations in which the sentence is true. A sentence is only a logical truth if it is true in *every* interpretation, and there are infinitely many interpretations. We need to reason about all of them, and we cannot do this by dealing with them one by one!

Sometimes, we can reason about all interpretations fairly easily. For example, we can offer a relatively simple argument that $Raa \leftrightarrow Raa$ is a logical truth:

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

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

Note another feature of the argument. Since the sentence in question contained no quantifiers, we did not need to think about how to interpret a and R; the point was just that, however we interpreted them, Raa would have some truth value or other. (We could ultimately have given the same argument concerning TFL sentences.)

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

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

This is quite longwinded, but, as things stand, there is no alternative. In order to show that a sentence is a logical truth, we must reason about *all* interpretations.

33.2 Other cases

Similar points hold of other cases too. Thus, we must reason about all interpretations if we want to show:

- that a sentence is a contradiction; for this requires that it is false in *every* interpretation.

- that two sentences are logically equivalent; for this requires that they have the same truth value in *every* interpretation.
- that some sentences are jointly inconsistent; for this requires that there is no interpretation in which all of those sentences are true together; i.e. that, in *every* interpretation, at least one of those sentences is false.
- that an argument is valid; for this requires that the conclusion is true in *every* interpretation where the premises are true.
- that some sentences entail another sentence.

The problem is that, with the tools available to you so far, reasoning about all interpretations is a serious challenge! Let's take just one more example. Here is an argument which is obviously valid:

$$\forall x(Hx \,\&\, Jx) \therefore \forall x Hx$$

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

> Consider an arbitrary interpretation in which the premise $\forall x(Hx \& Jx)$ is true. It follows that, however we expand the interpretation with a new name, for example c, $Hc \& Jc$ will be true in this new interpretation. Hc will, then, also be true in this new interpretation. But since this held for *any* way of expanding the interpretation, it must be that $\forall x Hx$ is true in the old interpretation. We've assumed nothing about the interpretation except that it was one in which $\forall x(Hx \,\&\, Jx)$ is true, so any interpretation in which $\forall x(Hx \,\&\, Jx)$ is true is one in which $\forall x Hx$ is true. The argument is valid!

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

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

	Yes	**No**
logical truth?	all interpretations	one counter-interpretation
contradiction?	all interpretations	one counter-interpretation
equivalent?	all interpretations	one counter-interpretation
consistent?	one interpretation	all interpretations
valid?	all interpretations	one counter-interpretation
entailment?	all interpretations	one counter-interpretation

This might usefully be compared with the table at the end of §14. The key difference resides in the fact that TFL concerns truth tables, whereas FOL concerns interpretations. This difference is deeply important, since each truth-table only ever has finitely many lines, so that a complete truth table is a relatively tractable object. By contrast, there are infinitely many interpretations for any given sentence(s), so that reasoning about all interpretations can be a deeply tricky business.

PART VII

Natural deduction for FOL

CHAPTER 34

Basic rules for FOL

FOL makes use of all of the connectives of TFL. So proofs in FOL will use all of the basic and derived rules from Part IV. We will also use the proof-theoretic notions (particularly, the symbol '$\vdash$') introduced there. However, we will also need some new basic rules to govern the quantifiers.

34.1 Universal elimination

Consider:

- Everyone is happy
- Therefore: Catrin is happy.

This is a valid argument. Generally, then, from the claim that everything is F, you can infer that any particular thing is F. You name it; it's F. So the following should be fine:

1	$\forall x Fx$	
2	Fa	$\forall$E 1

We obtained line 2 by dropping the universal quantifier and replacing 'x' with 'a'.

This isn't restricted to simple properties. Consider the following argument:

- Every cat is sleeping.
- Therefore: If Fluffy is a cat, then she is sleeping.

This is a valid argument. And it will be allowed by our rule ∀E:

1	$\forall x(Cx \to Sx)$	
2	$Cf \to Sf$	∀E 1

Note here that we have to replace two instances of 'x' with our name: 'f' or Fluffy. Indeed it would have been fine to do with any name. We can even do it with names we already have. Consider the following:

- Pavel owes everyone money.
- Therefore: Pavel owes himself money.

We we symbolise this as:

- $\forall x Opx$
- Therefore: Opp

This is valid: the premise says *everything* in the domain owes money to Pavel; and Pavel is something in the domain. So it implies that Pavel owes money to himself. A closely related sentence is:

1. Pavel owes money to everyone *else*.

and that will not be symbolised as $\forall x Oxx$; however, we do not yet have the resources to symbolise at. In §VIII we will introduce identity which will allow us to formalise it properly.

This argument is also directly allowed by our rule ∀E:

1	$\forall x Opx$	
2	Opp	∀E 1

We can now give our general rule using the notation from §30: Whenever you have a sentence $\forall x X(\ldots x \ldots x \ldots)$, for example $\forall x Fx$, $\forall x(Cx \to Sx)$, $\forall x Opx$; one can conclude that we have the sentence which is obtained by stripping of the quantifier and replacing the variable by a name, be it $a, b, c \ldots$. So we could derive Fa, $Cf \to Sf$ or Opp.

Here is the formal specification of the universal elimination rule (∀E):

$$
\begin{array}{r|ll}
m & \forall x X(\ldots x \ldots x \ldots) & \\
 & X(\ldots c \ldots c \ldots) & \forall\text{E } m
\end{array}
$$

The point is that you can obtain any *substitution instance* of a universally quantified formula: replace every instance of the quantified variable with any name you like.

I should emphasize that (as with every elimination rule) you can only apply the ∀E rule when the universal quantifier is the main logical operator. Thus the following is outright banned:

$$
\begin{array}{r|ll}
1 & \forall x Bx \rightarrow Bk & \\
\hline
2 & Bb \rightarrow Bk & \text{naughtily attempting to invoke } \forall\text{E } 1
\end{array}
$$

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

34.2 Existential introduction

The following argument is valid:

- Catrin is happy
- Therefore: Someone is happy.

This is the idea of our existential introduction rule: from the claim that some particular thing is F, you can infer that something is F:

$$
\begin{array}{r|ll}
1 & Fa & \\
\hline
2 & \exists x Fx & \exists\text{I } 1
\end{array}
$$

We obtained line 2 by replacing the name 'a' with the variable 'x' and adding $\exists x$ in front of the sentence. This will be permissible by our rule of ∃I.

This isn't restricted to simple properties.

- Bob is a money and knows sign language.
- Therefore: There is a monkey who knows sign language.

$$
\begin{array}{r|ll}
1 & Mb \,\&\, Sb & \\
\hline
2 & \exists x(Mx \,\&\, Sx) & \exists\text{I } 1
\end{array}
$$

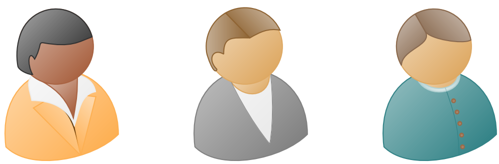
Or even

- Catrin is friends with someone who is friends with everyone.
- Therefore: Someone is friends with someone who is friends with everyone.

$$
\begin{array}{r|ll}
1 & \exists x(Fcx \,\&\, \forall y Fxy) & \\
\hline
2 & \exists z \exists x(Fzx \,\&\, \forall y Fxy) & \exists\text{I } 1
\end{array}
$$

We replaced the name, 'c' with the variable 'z' and added $\exists z$ at the beginning of the sentence.

This rule will now allow us to carefully work through our brain teaser from §18.2

> Three people are standing in a row looking at eachother.
>
> 
>
>
> Alice is happy. Charlie is not happy. Is there someone who is happy who is looking at someone who is not happy?

Answer: Yes.

We can formalise this argument as:

$$Lab,\ Lbc,\ Ha,\ \neg Hc \ \therefore\ \exists x \exists y(Hx \,\&\, (Lxy \,\&\, \neg Hx)).$$

And we can show it is valid. In §18.2 we wrote this in a pseudo-formal style.

1	Bob is either happy or he's not happy	
2	Suppose Bob is happy	
3	Then happy Bob is looking at unhappy Charlie	
4	So someone who is happy is looking at someone who is not happy.	
5	Suppose Bob is unhappy	
6	Then happy Alice is looking at unhappy Bob	
7	So someone who is happy is looking at someone who is not happy.	
8	Therefore, someone who is happy is looking at someone who is not happ	

We can now fill out the details of this to see that it's a formal proof:

1	Lab	
2	Lbc	
3	Ha	
4	$\neg Hc$	
5	$Hb \lor \neg Hb$	LEM
6	Hb	
7	$Lbc \,\&\, \neg Hc$	&I 2, 6
8	$Hb \,\&\, (Lbc \,\&\, \neg Hc)$	&I 7, 6
9	$\exists y(Hb \,\&\, (Lby \,\&\, \neg Hy))$	∃I 8
10	$\exists x \exists y(Hx \,\&\, (Lxy \,\&\, \neg Hx))$	∃I 9
11	$\neg Hb$	
12	$Lab \,\&\, \neg Hb$	&I 11, 1
13	$Ha \,\&\, (Lab \,\&\, \neg Hb)$	&I 3, 12
14	$\exists y(Ha \,\&\, (Lay \,\&\, \neg Hy))$	∃I 13
15	$\exists x \exists y(Hx \,\&\, (Lxy \,\&\, \neg Hx))$	∃I 14
16	$\exists x \exists y(Hx \,\&\, Lxy \,\&\, \neg Hx)$	∨E 5, 6–10, 11–15

Consider the following example:

- Narcissus loves himeself.
- Therefore: There is someone who loves Narcissus.

This is a valid argument. The formalised version, which will be allowed by our rule is:

$$
\begin{array}{r|ll}
1 & Lnn & \\
\hline
2 & \exists x Lxn & \exists\text{I } 1
\end{array}
$$

This shows us that we do not have to replace *all* instances of the name with the variable. Though of course we can if we wish: we could also deduce there is someone who loves himself.

To give our rule in general we need to introduce some new notation for this ability to replace just some of our instances of the name: If X is a sentence containing the name c, we can emphasize this by writing '$X(\ldots c \ldots c \ldots)$'. We will write '$X(\ldots x \ldots c \ldots)$' to indicate any formula obtained by replacing *some or all* of the instances of the name c with the variable x. Armed with this, our introduction rule is:

$$
\begin{array}{r|ll}
m & X(\ldots c \ldots c \ldots) & \\
 & \exists x X(\ldots x \ldots c \ldots) & \exists\text{I } m
\end{array}
$$

x must not occur in $X(\ldots c \ldots c \ldots)$

All the cases we've seen in this section follow this rule.

You might have noticed the additional constraint that's added to the rule. It is part of the rule; so if you are asked to write the rule ∃I you must include this constraint. However, you will not need to worry about it in practice. It's simply there to guarantee that applications of the rule yield sentences of FOL. If the rule were not there we would be allowed to argue as follows:

$$
\begin{array}{r|ll}
1 & \exists x Lnx & \\
\hline
2 & \exists x \exists x Lxx & \text{naughtily attempting to invoke } \exists\text{I } 1
\end{array}
$$

But this expression on line 2 contains clashing variables. It will not count as a sentence of FOL.

34.3 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 §24, we stipulated that domains in FOL must have at least one member. We then defined a logical truth (of FOL) as a sentence which is true in every interpretation. Since '$\exists x(Ax \lor \neg Ax)$' 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 $\forall$E.
- $Fa \vdash \exists x Fx$: after all, that was $\exists$I.
- the ability to copy-and-paste proofs together: after all, reasoning works by putting lots of little steps together into rather big chains.

If we get what we want on all three counts, then we have to countenance that $\forall x Fx \vdash \exists x Fx$. So, if we get what we want on all three counts, the proof system alone tells us that there is something rather than nothing. And if we refuse to accept that, then we have to surrender one of the three things that we want to hold on to!

Before we start thinking about which to surrender, we might want to ask how *much* of a cheat this is. Granted, it may make it harder to engage in theological debates about why there is something rather than nothing. But the rest of the time, we will get along just fine. So maybe we should just regard our proof system (and FOL, more generally) as having a very slightly limited purview. If we ever want to allow for the possibility of *nothing*, then we will have to cast around for a more complicated proof system. But for as long as we are content to ignore that possibility, our proof system is perfectly in order. (As, similarly, is the stipulation that every domain must contain at least one object.)

34.4 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 FOL, this process would never come to an end. So we could never apply that rule. We need to be a bit more cunning in coming up with our rule for introducing universal quantification.

Our cunning thought will be inspired by considering:

$$\forall x Fx \therefore \forall y Fy$$

This argument should *obviously* be valid. After all, alphabetical variation ought to be a matter of taste, and of no logical consequence. But how might our proof system reflect this? Suppose we begin a proof thus:

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 there 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	$X(\ldots c \ldots c \ldots)$	
	$\forall x X(\ldots x \ldots x \ldots)$	$\forall$I m

c must not occur in any undischarged assumption
x must not occur in $X(\ldots c \ldots c \ldots)$

A crucial aspect of this rule, though, is bound up in the first constraint. This constraint ensures that we are always reasoning at a sufficiently general level. To see the constraint in action, consider this terrible argument:

> Everyone loves Kylie Minogue; therefore everyone loves themselves.

We might symbolize this obviously invalid inference pattern as:

$$\forall x Lxk \therefore \forall x Lxx$$

Now, suppose we tried to offer a proof that vindicates this argument:

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

This is not allowed, because 'k' occurred already in an undischarged assumption, namely, on line 1. The crucial point is that, if we have made any assumptions about the object we are working with, then we are not reasoning generally enough to license $\forall$I.

Although the name may not occur in any *undischarged* assumption, it may occur as a discharged assumption. That is, it may occur in a subproof that we have already closed. For example:

1	Gd	
2	Gd	R 1
3	$Gd \rightarrow Gd$	$\rightarrow$I 1–2
4	$\forall z(Gz \rightarrow Gz)$	$\forall$I 3

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

34.5 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:

$$
\begin{array}{r|l l}
1 & \exists x Fx & \\
2 & \underline{\forall x(Fx \to Gx)} & \\
3 & \quad \underline{Fo} & \\
4 & \quad Fo \to Go & \forall\text{E } 2 \\
5 & \quad Go & \to\text{E } 4, 3 \\
6 & \quad \exists x Gx & \exists\text{I } 5 \\
7 & \exists x Gx & \exists\text{E } 1, 3\text{–}6
\end{array}
$$

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 '$\exists xGx$'. 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 ($\exists$E):

$$
\begin{array}{r|l l}
m & \exists x \mathcal{X}(\ldots x \ldots x \ldots) & \\
i & \quad \underline{\mathcal{X}(\ldots c \ldots c \ldots)} & \\
 & \quad \vdots & \\
j & \quad \mathcal{Y} & \\
 & \mathcal{Y} & \exists\text{E } m, i\text{–}j
\end{array}
$$

c must not occur in any assumption undischarged before line i
c must not occur in $\exists x\mathcal{X}(\ldots x \ldots x \ldots)$
c must not occur in $\mathcal{Y}$

As with universal introduction, the constraints are extremely important. To see why, consider the following terrible argument:

> Tim Button is a lecturer. There is someone who is not a lecturer. So Tim Button is both a lecturer and not a lecturer.

We might symbolize this obviously invalid inference pattern as follows:

$$Lb, \exists x \neg Lx \therefore Lb \mathbin{\&} \neg Lb$$

Now, suppose we tried to offer a proof that vindicates this argument:

$$\begin{array}{r|l l}
1 & Lb & \\
2 & \exists x \neg Lx & \\
\hline
3 & \quad \vert\ \neg Lb & \\
4 & \quad \vert\ Lb \mathbin{\&} \neg Lb & \text{\&I 1, 3} \\
5 & Lb \mathbin{\&} \neg Lb & \text{naughtily attempting to invoke } \exists\text{E } 2, 3\text{–}4
\end{array}$$

The last line of the proof is not allowed. The name that we used in our substitution instance for '$\exists x \neg Lx$' on line 3, namely 'b', occurs in line 4. The following proof would be no better:

$$\begin{array}{r|l l}
1 & Lb & \\
2 & \exists x \neg Lx & \\
\hline
3 & \quad \vert\ \neg Lb & \\
4 & \quad \vert\ Lb \mathbin{\&} \neg Lb & \text{\&I 1, 3} \\
5 & \quad \vert\ \exists x(Lx \mathbin{\&} \neg Lx) & \exists\text{I 4} \\
6 & \exists x(Lx \mathbin{\&} \neg Lx) & \text{naughtily attempting to invoke } \exists\text{E } 2, 3\text{–}5
\end{array}$$

The last line of the proof would still not be allowed. For the name that we used in our substitution instance for '$\exists x \neg Lx$', namely 'b', occurs in an undischarged assumption, namely line 1.

The moral of the story is this. *If you want to squeeze information out of an existential quantifier, choose a new name for your substitution instance.* That way, you can guarantee that you meet all the constraints on the rule for ∃E.

CHAPTER 35

Conversion of quantifiers

In this section, we will add some additional rules to the basic rules of the previous section. These govern the interaction of quantifiers and negation.

In §24, we noted that $\neg\exists x X$ is logically equivalent to $\forall x \neg X$. We will add some rules to our proof system that govern this. In particular, we add:

m	$\forall x \neg X$	
	$\neg\exists x X$	CQ m

and

m	$\neg\exists x X$	
	$\forall x \neg X$	CQ m

Equally, we add:

$$
\begin{array}{r|ll}
m & \exists x \neg X & \\
 & \neg \forall x X & \text{CQ } m
\end{array}
$$

and

$$
\begin{array}{r|ll}
m & \neg \forall x X & \\
 & \exists x \neg X & \text{CQ } m
\end{array}
$$

CHAPTER 36

Derived rules

As in the case of TFL, we first introduced some rules for FOL as basic (in §34), and then added some further rules for conversion of quantifiers (in §35). In fact, the CQ rules should be regarded as *derived* rules, for they can be derived from the *basic* rules of §34. (The point here is as in §21.) Here is a justification for the first CQ rule:

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

Here is a justification of the third CQ rule:

$$
\begin{array}{lllll}
1 & \exists x \neg Ax & & & \\
2 & \quad \forall x Ax & & & \\
3 & \quad\quad \neg Ac & & & \\
4 & \quad\quad Ac & & & \forall\mathrm{E}\ 2 \\
5 & \quad\quad \bot & & & \bot\mathrm{I}\ 4, 3 \\
6 & \quad \bot & & & \exists\mathrm{E}\ 1, 3\text{–}5 \\
7 & \neg\forall x Ax & & & \neg\mathrm{I}\ 2\text{–}6
\end{array}
$$

This explains why the CQ rules can be treated as derived. Similar justifications can be offered for the other two CQ rules.

CHAPTER 37

Proof-theoretic and semantic concepts

We have used two different turnstiles in this book. This:

$$X_1, X_2, \ldots, X_n \vdash C$$

means that there is some proof which starts with assumptions $X_1, X_2, \ldots, X_n$ and ends with C (and no undischarged assumptions other than $X_1, X_2, \ldots, X_n$). This is a *proof-theoretic notion.*

By contrast, this:

$$X_1, X_2, \ldots, X_n \vDash C$$

means that there is no valuation (or interpretation) which makes all of $X_1, X_2, \ldots, X_n$ true and makes C false. This concerns assignments of truth and falsity to sentences. It is a *semantic notion.*

It cannot be emphasized enough that these are different notions. But we can emphasize it a bit more: *They are different notions.*

Once you have internalised this point, continue reading.

Although our semantic and proof-theoretic notions are different, there is a deep connection between them. To explain this connection,we will start by considering the relationship between logical truths and theorems.

To show that a sentence is a theorem, you need only produce a proof. Granted, it may be hard to produce a twenty line proof, but it is not so hard to check each line of the proof and confirm that it is legitimate; and if each line of the proof individually is legitimate, then the whole proof is legitimate. Showing that a sentence is a logical truth, though, requires reasoning about all possible interpretations. Given a choice between showing that a sentence is a theorem and showing that it is a logical truth, it would be easier to show that it is a theorem.

Contrawise, to show that a sentence is *not* a theorem is hard. We would need to reason about all (possible) proofs. That is very difficult. However, to show that a sentence is not a logical truth, you need only construct an interpretation in which the sentence is false. Granted, it may be hard to come up with the interpretation; but once you have done so, it is relatively straightforward to check what truth value it assigns to a sentence. Given a choice between showing that a sentence is not a theorem and showing that it is not a logical truth, it would be easier to show that it is not a logical truth.

Fortunately, *a sentence is a theorem if and only if it is a logical truth.* As a result, if we provide a proof of X on no assumptions, and thus show that X is a theorem, we can legitimately infer that X is a logical truth; i.e., $\vDash X$. Similarly, if we construct an interpretation in which X is false and thus show that it is not a logical truth, it follows that X is not a theorem.

More generally, we have the following powerful result:

$$X_1, X_2, \ldots, X_n \vdash Y \textbf{ iff } X_1, X_2, \ldots, X_n \vDash Y$$

This shows that, whilst provability and entailment are *different* notions, they are extensionally equivalent. As such:

- An argument is *valid* iff *the conclusion can be proved from the premises.*
- Two sentences are *logically equivalent* iff they are *provably equivalent.*
- Sentences are *provably consistent* iff they are *not provably inconsistent.*

For this reason, you can pick and choose when to think in terms of proofs and when to think in terms of valuations/interpretations, doing whichever is easier for a given task. The table on the next page summarises which is (usually) easier.

It is intuitive that provability and semantic entailment should agree. But—let us repeat this—do not be fooled by the similarity of the symbols '$\vDash$' and '$\vdash$'. These two symbols have very different meanings. The fact that provability and semantic entailment agree is not an easy result to come by.

In fact, demonstrating that provability and semantic entailment agree is, very decisively, the point at which introductory logic becomes intermediate logic.

	Yes	**No**
Is X a **logical truth**?	give a proof which shows $\vdash X$	give an interpretation in which X is false
Is X a **contradiction**?	give a proof which shows $\vdash \neg X$	give an interpretation in which X is true
Are X and Y **equivalent**?	give two proofs, one for $X \vdash Y$ and one for $Y \vdash X$	give an interpretation in which X and Y have different truth values
Are $X_1, X_2, \ldots, X_n$ **jointly consistent**?	give an interpretation in which all of $X_1, X_2, \ldots, X_n$ are true	prove a contradiction from assumptions $X_1, X_2, \ldots, X_n$
Is $X_1, X_2, \ldots, X_n \therefore C$ **valid**?	give a proof with assumptions $X_1, X_2, \ldots, X_n$ and concluding with C	give an interpretation in which each of $X_1, X_2, \ldots, X_n$ is true and C is false

PART VIII

Identity

CHAPTER 38

Identity

Consider this sentence:

1. Pavel owes money to everyone

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

Oxy: $\underline{\qquad}_x$ owes money to $\underline{\qquad}_y$
p: Pavel

we can symbolize sentence 1 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:

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

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

38.1 Adding identity

The new symbol we add is '='. This is a symbol that we can use for *identity*.

We will then be able symbolise

5. Clark Kent is Superman.

as $k = s$, using the symbolisation key

k: Clark Kent
s: Superman

This will also be a symbolisations of paraphrases of 5:

6. Clark Kent and Superman are the same person.
7. Clark Kent is identical to Superman.

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

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

8. No one besides Pavel owes money to Hikaru.
9. Only Pavel owes Hikaru money.

Let 'h' name Hikaru. Sentence 8 can be paraphrased as, 'No one who is not Pavel owes money to Hikaru'. This can be symbolized by '$\neg\exists x(\neg x = p \,\&\, Oxh)$'. Equally, sentence 8 can be paraphrased as 'for all x, if x owes money to Hikaru, then x is Pavel'. It can then be symbolized as '$\forall x(Oxh \rightarrow x = p)$'.

Sentence 9 can be treated similarly, but there is one subtlety here. Do either sentence 8 or 9 entail that Pavel himself owes money to Hikaru?

38.2 There are at least…

We will now look at more that we can do armed with our new identity symbol. We can also use identity to say how many things there are of a particular kind. For example, consider these sentences:

10. There is at least one apple
11. There are at least two apples
12. There are at least three apples

We will use the symbolization key:

Ax: $\underline{\hspace{3em}}_x$ is an apple

Sentence 10 does not require identity. It can be adequately symbolized by '$\exists x Ax$': There is an apple; perhaps many, but at least one.

It might be tempting to also symbolize sentence 11 without identity. Yet consider the sentence '$\exists x \exists y(Ax \,\&\, Ay)$'. Roughly, this says that there is some apple x in the domain and some apple y in the domain. Since nothing precludes these from being one and the same apple, this would be true even if there were only one apple. In order to make sure that we are dealing with *different* apples, we need an identity predicate. Sentence 11 needs to say that the two apples that exist are not identical, so it can be symbolized by '$\exists x \exists y((Ax \,\&\, Ay) \,\&\, \neg x = y)$'.

Sentence 12 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: '$\exists x \exists y \exists z[((Ax \,\&\, Ay) \,\&\, Az) \,\&\, ((\neg x = y \,\&\, \neg y = z) \,\&\, \neg x = z)]$'.

38.3 There are at most. . .

Now consider these sentences:

13. There is at most one apple
14. There are at most two apples

Sentence 13 can be paraphrased as, 'It is not the case that there are at least *two* apples'. This is just the negation of sentence 11:

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

But sentence 13 can also be approached in another way. It means that if you pick out an object and it's an apple, and then you pick out an object and it's also an apple, you must have picked out the same object both times. With this in mind, it can be symbolized by

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

The two sentences will turn out to be logically equivalent.

In a similar way, sentence 14 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:

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

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

$$\forall x \forall y \forall z \big[(Ax \mathbin{\&} Ay \mathbin{\&} Az) \rightarrow (x = y \lor x = z \lor y = z)\big]$$

38.4 There are exactly...

We can now consider precise statements, like:

15. There is exactly one apple.
16. There are exactly two apples.
17. There are exactly three apples.

Sentence 15 can be paraphrased as, 'There is *at least* one apple and there is *at most* one apple'. This is just the conjunction of sentence 10 and sentence 13. So we can offer:

$$\exists x Ax \mathbin{\&} \forall x \forall y \big[(Ax \mathbin{\&} Ay) \rightarrow x = y\big]$$

But it is perhaps more straightforward to paraphrase sentence 15 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:

$$\exists x \big[Ax \mathbin{\&} \forall y (Ay \rightarrow x = y)\big]$$

Similarly, sentence 16 may be paraphrased as, 'There are *at least* two apples, and there are *at most* two apples'. Thus we could offer

$$\exists x \exists y ((Ax \mathbin{\&} Ay) \mathbin{\&} \neg x = y) \mathbin{\&} \\ \forall x \forall y \forall z \big[((Ax \mathbin{\&} Ay) \mathbin{\&} Az) \rightarrow ((x = y \lor x = z) \lor y = z)\big]$$

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:

$$\exists x \exists y \big[((Ax \mathbin{\&} Ay) \mathbin{\&} \neg x = y) \mathbin{\&} \forall z (Az \rightarrow (x = z \lor y = z)\big]$$

Finally, consider these sentence:

18. There are exactly two things
19. There are exactly two objects

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

$$\exists x \exists y \neg\, x = y \;\&\; \neg \exists x \exists y \exists z((\neg\, x = y \;\&\; \neg\, y = z) \;\&\; \neg\, x = z)$$
$$\exists x \exists y \big[\neg\, x = y \;\&\; \forall z(x = z \lor y = z)\big]$$

CHAPTER 39

Definite descriptions

Consider sentences like:

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

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

39.1 Treating definite descriptions as terms

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

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

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

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

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

39.2 Russell's analysis

Bertrand Russell offered an analysis of definite descriptions. Very briefly put, he observed that, when we say 'the F' in the context of a definite description, our aim is to pick out the *one and only* thing that is F (in the appropriate context). Thus Russell analysed the notion of a definite description as follows:[1]

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

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

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

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

$$\exists x Fx \mathbin{\&} \forall x \forall y((Fx \mathbin{\&} Fy) \rightarrow x = y) \mathbin{\&} \forall x(Fx \rightarrow Gx)$$

In fact, we could express the same point rather more crisply, by recognizing that the first two conjuncts just amount to the claim that there

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

is *exactly* one F, and that the last conjunct tells us that that object is F. So, equivalently, we could offer:

$$\exists x\big[(Fx \mathbin{\&} \forall y(Fy \rightarrow x = y)) \mathbin{\&} Gx\big]$$

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

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

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

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

$$\exists x \exists y\big(\big[Tx \mathbin{\&} \forall z(Tz \rightarrow x = z)\big] \mathbin{\&} \big[Dy \mathbin{\&} \forall z(Dz \rightarrow y = z)\big] \mathbin{\&} x = y\big)$$

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

39.3 Empty definite descriptions

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

France has no king at present. Now, if we were to introduce a name, 'k', to name the present King of France, then everything would go wrong: remember from §24 that a name must always pick out some object in the domain, and whatever we choose as our domain, it will contain no present kings of France.

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

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

4. I am dating the present king of France.

Using the following symbolization key:

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

a: Alex
Kx: $\_\_\_\_\__x$ is a present king of France
Dxy: $\_\_\_\_\__x$ is dating $\_\_\_\_\__y$

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

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

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

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

39.4 The adequacy of Russell's analysis

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

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

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

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

[3] P.F. Strawson, 'On Referring', 1950, *Mind* 59, pp. 320–34.

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

7. The gin-drinker is very tall!

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

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

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

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

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

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

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

CHAPTER 40

Semantics for FOL with identity

FOL with identity extends FOL as we presented it earlier.

40.1 Sentences of FOL with identity

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

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

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

This is added to the other clauses of what it is to be a sentence as they were given in §27

We can now see that $\forall x(\neg\, x = p \rightarrow Opx)$ is a sentence, as it could be constructed as follows:

$$\forall x(\neg x = p \rightarrow Opx)$$

$$(\neg c = p \rightarrow Opc)$$

$$\begin{array}{cc} \neg c = p & Opc \\ c = p & \end{array}$$

Be aware that when you see $\neg a = b$ the negation is attached to the whole sentence $a = b$, not to a. So, you should not write $a = \neg b$. This is not a sentence. Sometimes you might see $a \neq b$, and this is short hand for $\neg a = b$.

Sometimes you might come across cases where you might feel tempted to write something like $a = \neg b$. Such temptation should be avoided. For example, in our semi-formalised English we might say 'for all x, if x is not p, then Ax'. But note that this should be $\forall x(\neg x = p \rightarrow Ax)$, and should not be written with $x = \neg p$.

40.2 Semantics for identity

Now that we have added the identity symbol to FOL, we simply need to expand our notion of truth from Part VI to also account for atomic sentences like $a = b$. Our clause that we add is:

> $a = b$ is true **iff** a and b name the same object in that interpretation.

So on our go-to interpretation from Part VI,

domain: all people born before 2000CE
a: Aristotle
b: Beyoncé
Px: $\_\_\_\_\__x$ is a philosopher
Rxy: $\_\_\_\_\__x$ was born before $\_\_\_\_\__y$

we have that $a = b$ is false: Aristotle and Beyoncé are different people, so 'a' and 'b' name different objects.

Consider the following interpretation

domain: All celestial bodies

e: The evening star
m: The morning star

It turns out that The Morning Star is *the same object as* The Evening Star: they are names for Venus. So here we have two names for the same object. We thus have $e = m$ is true on this interpretation.

Identity becomes particularly useful when we have quantifiers. Suppose we have

domain: Alfred, Billy, Carys
a: Alfred
b: Billy
c: Carys

Remember, to check if $\exists x X(\ldots x \ldots x \ldots)$ is true we first add a new name, let's use d, and we see if there is some way of extending the domain so that $X(\ldots d \ldots d \ldots)$ is true. So, let's see if $\exists x\, x = a$ is true: we add a new name d and consider the extended interpretation with

d: Alfred

Then $d = a$ is true on this interpretation. So there is some way of interpreting 'd' where $d = a$ is true; and thus $\exists x\, x = a$ is true.

On this interpretation we can also see that $\forall x(x = a \vee x = b \vee x = c)$ is true. Why? We add a new name 'd'. There are three ways we can extend our original interpretation:

1. d: Alfred
2. d: Billy
3. d: Carys

On the first of these, $d = a$ is true, on the second $d = b$ is true, and on the third $d = c$ is true. So on each of these interpretations, $d = a \vee d = b \vee d = c$ is true, and thus $\forall x(x = a \vee x = b \vee x = c)$.

CHAPTER 41

Rules for identity

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

$$Aa \therefore Ab$$
$$Ab \therefore Aa$$
$$Raa \therefore Rbb$$
$$Raa \therefore Rab$$
$$Rca \therefore Rcb$$
$$\forall x Rxa \therefore \forall x Rxb$$

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

$$
\begin{array}{r|ll}
m & a = b & \\
n & X(\ldots a \ldots a \ldots) & \\
 & X(\ldots b \ldots a \ldots) & {=}\mathrm{E}\ m,\ n
\end{array}
$$

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

$$
\begin{array}{r|ll}
m & a = b & \\
n & X(\ldots b \ldots b \ldots) & \\
 & X(\ldots a \ldots b \ldots) & {=}\mathrm{E}\ m,\ n
\end{array}
$$

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

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

To bring this out, consider the following interpretation:

domain: P.D. Magnus, Tim Button
- a: P.D. Magnus
- b: Tim Button
- For every primitive predicate we care to consider, that predicate is true of *nothing*.

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

Since we are not subscribing to the thesis of identity of indiscernibles, no matter how much you learn about two objects, we cannot prove that they are identical. That is unless, of course, you learn that the two objects are, in fact, identical, but then the proof will hardly be very illuminating.

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

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

$$\begin{array}{|ll} c = c & \quad =\text{I} \end{array}$$

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

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

$$\begin{array}{r|l|l}
1 & \quad a = b & \\
\hline
2 & \quad a = a & =\text{I} \\
3 & \quad b = a & =\text{E } 1, 2 \\
4 & a = b \rightarrow b = a & \rightarrow\text{I } 1\text{–}3 \\
5 & \forall y(a = y \rightarrow y = a) & \forall\text{I } 4 \\
6 & \forall x \forall y(x = y \rightarrow y = x) & \forall\text{I } 5
\end{array}$$

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

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

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

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

PART IX

Metatheory (Non-examinable)

CHAPTER 42

Normal forms

42.1 Disjunctive Normal Form

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

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

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

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

$$A$$
$$(A \,\&\, \neg B \,\&\, C)$$
$$(A \,\&\, B) \lor (A \,\&\, \neg B)$$
$$(A \,\&\, B) \lor (A \,\&\, B \,\&\, C \,\&\, \neg D \,\&\, \neg E)$$
$$A \lor (C \,\&\, \neg P_{234} \,\&\, P_{233} \,\&\, Q) \lor \neg B$$

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

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

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

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

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

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

42.2 Proof of DNF Theorem via truth tables

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

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

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

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

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

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

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

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

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

Pick any arbitrary sentence, S, and let $A_1, \ldots, A_n$ be the atomic sentences that occur in S. To obtain a sentence in DNF that is logically equivalent S, we consider S's truth table. There are two cases to consider:

1. *S is false on every line of its truth table.* Then, S is a contradiction. In that case, the contradiction $(A_1 \& \neg A_1)$ is in DNF and logically equivalent to S.

2. *S is true on at least one line of its truth table.* For each line i of the truth table, let B_i be a conjunction of the form

$$(\pm A_1 \wedge \ldots \wedge \pm A_n)$$

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

$$A_m \text{ is a conjunct of } B_i \textit{ iff } A_m \text{ is true on line } i$$
$$\neg A_m \text{ is a conjunct of } B_i \textit{ iff } A_m \text{ is false on line } i$$

Given these rules, B_i is true on (and only on) line i of the truth table which considers all possible valuations of $A_1, \ldots, A_n$ (i.e. S's truth table).

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

$$B_{i_1} \lor B_{i_2} \lor \ldots \lor B_{i_m}$$

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

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

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

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

42.3 Conjunctive Normal Form

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

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

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

Generally, then, a sentence in CNF looks like this

$$(\pm\mathcal{A}_1 \lor \ldots \lor \pm\mathcal{A}_i) \land (\pm\mathcal{A}_{i+1} \lor \ldots \lor \pm\mathcal{A}_j) \land \ldots \land (\pm\mathcal{A}_{m+1} \lor \ldots \lor \pm\mathcal{A}_n)$$

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

We can now prove another normal form theorem:

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

Given a TFL sentence, S, we begin by writing down the complete truth table for S.

If S is *true* on every line of the truth table, then S and $(A_1 \vee \neg A_1)$ are logically equivalent.

If S is *false* on at least one line of the truth table then, for every line on the truth table where S is false, write down a disjunction $(\pm A_1 \vee \ldots \vee \pm A_n)$ which is *false* on (and only on) that line. Let C be the conjunction of all of these disjuncts; by construction, C is in CNF and S and C are logically equivalent.

CHAPTER 43

Expressive Adequacy

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

$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\heartsuit(\mathcal{A},\mathcal{B},\mathcal{C})$
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	F

Probably this new connective would not correspond with any natural English expression (at least not in the way that '&' corresponds with 'and'). But a question arises: if we wanted to employ a connective with this characteristic truth table, must we add a *new* connective to TFL? Or can we get by with the connectives we *already have*?

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

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

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

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

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

$$(A \,\&\, B \,\&\, \neg C) \vee (A \,\&\, \neg B \,\&\, C) \vee (\neg A \,\&\, B \,\&\, \neg C)$$

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

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

$$(A \,\&\, B) \quad \text{and} \quad \neg(\neg A \vee \neg B)$$

are logically equivalent.

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

$$(A \vee B) \quad \text{and} \quad \neg(\neg A \,\&\, \neg B)$$

are logically equivalent.

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

$$(A \vee B) \quad \text{and} \quad (\neg A \rightarrow B)$$
$$(A \,\&\, B) \quad \text{and} \quad \neg(A \rightarrow \neg B)$$

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

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

43.1 Individually expressively adequate connectives

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

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

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

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

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

The Expressive Adequacy Theorem tells us that '$\neg$' and '$\vee$' are jointly expressively adequate. So it suffices to show that, given any scheme which contains only those two connectives, we can rewrite it as a logically equivalent scheme which contains only '$\uparrow$'. As in the proof

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

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

of the subsidiary cases of the Expressive Adequacy Theorem, then, we simply apply the following equivalences:

$$\begin{array}{rcl} \neg A & \text{and} & (A \uparrow A) \\ (A \vee B) & \text{and} & ((A \uparrow A) \uparrow (B \uparrow B)) \end{array}$$

to the Subsidiary Result 1.

Similarly, we can consider the connective '↓':

A	B	$A \downarrow B$
T	T	F
T	F	F
F	T	F
F	F	T

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

> '↓' is expressively adequate all by itself.

As in the previous result for ↑, although invoking the equivalences:

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

and Subsidiary Result 2.

43.2 Failures of expressive adequacy

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

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

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

A	$\neg A$
T	F
F	T

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

> '$\vee$', '&', '$\to$', and '$\leftrightarrow$' are not expressively adequate by themselves.

In fact, the following is true:

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

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

CHAPTER 44

Soundness

In this chapter we relate TFL's semantics to its natural deduction *proof system* (as defined in Part IV). We will prove that the formal proof system is safe: you can only prove sentences from premises from which they actually follow. Intuitively, a formal proof system is sound iff it does not allow you to prove any invalid arguments. This is obviously a highly desirable property. It tells us that our proof system will never lead us astray. Indeed, if our proof system were not sound, then we would not be able to trust our proofs. The aim of this chapter is to prove that our proof system is sound.

Let's make the idea more precise. We'll abbreviate a list of sentences using the greek letter Γ ('gamma'). A formal proof system is SOUND (relative to a given semantics) *iff*, whenever there is a formal proof of $\mathcal{Z}$ from assumptions among Γ, then Γ genuinely entails $\mathcal{Z}$ (given that semantics). Otherwise put, to prove that TFL's proof system is sound, we need to prove the following

> **Soundness Theorem.** For any sentences Γ and $\mathcal{Z}$: if $\Gamma \vdash \mathcal{Z}$, then $\Gamma \vDash \mathcal{Z}$

To prove this, we will check each of the rules of TFL's proof system individually. We want to show that no application of those rules ever leads us astray. Since a proof just involves repeated application of those rules, this will show that no proof ever leads us astray. Or at least, that is the general idea.

To begin with, we must make the idea of 'leading us astray' more precise. Say that a line of a proof is SHINY iff the assumptions on which that line depends tautologically entail the sentence on that line. The word 'shiny' is not standard among logicians, but it will help us with our discussions. To illustrate the idea, consider the following:

$$
\begin{array}{r|l|l}
1 & F \to (G \,\&\, H) & \\
\hline
2 & \quad F & \\
3 & \quad G \,\&\, H & \to\text{E } 1, 2 \\
4 & \quad G & \&\text{E } 3 \\
5 & F \to G & \to\text{I } 2\text{–}4
\end{array}
$$

Line 1 is shiny iff $F \to (G \,\&\, H) \vDash F \to (G \,\&\, H)$. You should be easily convinced that line 1 is, indeed, shiny! Similarly, line 4 is shiny iff $F \to (G \,\&\, H), F \vDash G$. Again, it is easy to check that line 4 is shiny. As is every line in this TFL-proof. We want to show that this is no coincidence. That is, we want to prove:

Shininess Lemma. Every line of every TFL-proof is shiny.

Then we will know that we have never gone astray, on any line of a proof. Indeed, given the Shininess Lemma, it will be easy to prove the Soundness Theorem:

Proof. Suppose $\Gamma \vdash Z$. Then there is a TFL-proof, with Z appearing on its last line, whose only undischarged assumptions are among Γ. The Shininess Lemma tells us that every line on every TFL-proof is shiny. So this last line is shiny, i.e. $\Gamma \vDash Z$. QED

It remains to prove the Shininess Lemma.

To do this, we observe that every line of any TFL-proof is obtained by applying some rule. So what we want to show is that no application of a rule of TFL's proof system will lead us astray. More precisely, say that a rule of inference is RULE-SOUND *iff* for all TFL-proofs, if we obtain a line on a TFL-proof by applying that rule, and every earlier line in the TFL-proof is shiny, then our new line is also shiny. What we need to show is that *every* rule in TFL's proof system is rule-sound.

We will do this in the next section. But having demonstrated the rule-soundness of every rule, the Shininess Lemma will follow immediately:

Proof. Fix any line, line n, on any TFL-proof. The sentence written on line n must be obtained using a formal inference rule which is rule-sound. This is to say that, if every earlier line is shiny, then line n itself is shiny. Hence, by strong induction on the length of TFL-proofs, every line of every TFL-proof is shiny. QED

Note that this proof appeals to a principle of strong induction on the length of TFL-proofs. This is the first time we have seen that principle, and you should pause to confirm that it is, indeed, justified.

It remains to show that every rule is rule-sound. This is not difficult, but it is time-consuming, since we need to check each rule individually, and TFL's proof system has plenty of rules! To speed up the process marginally, we will introduce a convenient abbreviation: 'Δ_i' ('delta') will abbreviate the assumptions (if any) on which line i depends in our TFL-proof (context will indicate which TFL-proof we have in mind).

> Introducing an assumption is rule-sound.

If X is introduced as an assumption on line n, then X is among Δ_n, and so $\Delta_n \vDash X$.

> &I is rule-sound.

Proof. Consider any application of &I in any TFL-proof, i.e., something like:

i	X	
j	Y	
n	$X \,\&\, Y$	&I i, j

To show that &I is rule-sound, we assume that every line before line n is shiny; and we aim to show that line n is shiny, i.e. that $\Delta_n \vDash X \,\&\, Y$.

So, let v be any valuation that makes all of Δ_n true.

We first show that v makes X true. To prove this, note that all of Δ_i are among Δ_n. By hypothesis, line i is shiny. So any valuation that makes all of Δ_i true makes X true. Since v makes all of Δ_i true, it makes X true too.

We can similarly see that v makes Y true.

So v makes X true and v makes Y true. Consequently, v makes $X \,\&\, Y$ true. So any valuation that makes all of the sentences among Δ_n true also makes $X \,\&\, Y$ true. That is: line n is shiny. QED

All of the remaining lemmas establishing rule-soundness will have, essentially, the same structure as this one did.

&E is rule-sound.

Proof. Assume that every line before line n on some TFL-proof is shiny, and that &E is used on line n. So the situation is:

$$
\begin{array}{l|ll}
i & X \,\&\, Y & \\
n & X & \&\mathrm{E}\ i
\end{array}
$$

(or perhaps with Y on line n instead; but similar reasoning will apply in that case). Let v be any valuation that makes all of Δ_n true. Note that all of Δ_i are among Δ_n. By hypothesis, line i is shiny. So any valuation that makes all of Δ_i true makes $X \,\&\, Y$ true. So v makes $X \,\&\, Y$ true, and hence makes X true. So $\Delta_n \vDash X$. QED

$\vee$I is rule-sound.

We leave this as an exercise.

$\vee$E is rule-sound.

Proof. Assume that every line before line n on some TFL-proof is shiny, and that &E is used on line n. So the situation is:

$$
\begin{array}{l|l|ll}
m & X \vee Y & & \\
i & & \underline{X} & \\
j & & Z & \\
k & & \underline{Y} & \\
l & & Z & \\
n & Z & & \vee\mathrm{E}\ m,\ i\text{–}j,\ k\text{–}l
\end{array}
$$

Let v be any valuation that makes all of Δ_n true. Note that all of Δ_m are among Δ_n. By hypothesis, line m is shiny. So any valuation that makes Δ_n true makes $X \vee Y$ true. So in particular, v makes $X \vee Y$ true, and hence either v makes X true, or v makes Y true. We now reason through these two cases:

Case 1: v makes X true. All of Δ_i are among Δ_n, with the possible exception of X. Since v makes all of Δ_n true, and also makes X true, v makes all of Δ_i true. Now, by assumption, line j is shiny; so $\Delta_j \vDash Z$. But the sentences Δ_i are just the sentences Δ_j, so $\Delta_i \vDash Z$. So, any valuation that makes all of Δ_i true makes Z true. But v is just such a valuation. So v makes Z true.

Case 2: v makes Y true. Reasoning in exactly the same way, considering lines k and l, v makes Z true.

Either way, v makes Z true. So $\Delta_n \vDash Z$. QED

¬E is rule-sound.

Proof. Assume that every line before line n on some TFL-proof is shiny, and that ¬E is used on line n. So the situation is:

$$\begin{array}{r|ll} i & X & \\ j & \neg X & \\ n & \bot & \bot\text{I } i, j \end{array}$$

Note that all of Δ_i and all of Δ_j are among Δ_n. By hypothesis, lines i and j are shiny. So any valuation which makes all of Δ_n true would have to make both X and $\neg X$ true. But no valuation can do that. So no valuation makes all of Δ_n true. So $\Delta_n \vDash \bot$, vacuously. QED

⊥I is rule-sound.

We leave this as an exercise.

¬I is rule-sound.

Proof. Assume that every line before line n on some TFL-proof is shiny, and that ¬I is used on line n. So the situation is:

$$\begin{array}{r|l|ll} i & & \underline{X} & \\ j & & \bot & \\ n & \neg X & & \neg\text{I } i\text{–}j \end{array}$$

Let v be any valuation that makes all of Δ_n true. Note that all of Δ_n are among Δ_i, with the possible exception of X itself. By hypothesis, line j is shiny. But no valuation can make '$\bot$' true, so no valuation can make all of Δ_j true. Since the sentences Δ_i are just the sentences Δ_j, no valuation can make all of Δ_i true. Since v makes all of Δ_n true, it must therefore make X false, and so make $\neg X$ true. So $\Delta_n \vDash \neg X$. QED

$\neg$I, $\bot$E, $\to$I, $\to$E, $\leftrightarrow$I, and $\leftrightarrow$E are all rule-sound.

We leave these as exercises.

This establishes that all the basic rules of our proof system are rule-sound. Finally, we show:

All of the derived rules of our proof system are rule-sound.

Proof. Suppose that we used a derived rule to obtain some sentence, X, on line n of some TFL-proof, and that every earlier line is shiny. Every use of a derived rule can be replaced (at the cost of long-windedness) with multiple uses of basic rules. That is to say, we could have used basic rules to write X on some line $n + k$, without introducing any further assumptions. So, applying our individual results that all basic rules are rule-sound several times ($k+1$ times, in fact), we can see that line $n + k$ is shiny. Hence the derived rule is rule-sound. QED

And that's that! We have shown that every rule—basic or otherwise—is rule-sound, which is all that we required to establish the Shininess Lemma, and hence the Soundness Theorem.

But it might help to round off this chapter if we repeat my informal explanation of what we have done. A formal proof is just a sequence—of arbitrary length—of applications of rules. We have shown that any application of any rule will not lead you astray. It follows (by induction)that no formal proof will lead you astray. That is: our proof system is sound.

PART X

Modal logic (non-examinable)

CHAPTER 45

Introducing modal logic

Modal logic (ML) is the logic of *modalities*, ways in which a statement can be true. *Necessity* and *possibility* are two such modalities: a statement can be true, but it can also be necessarily true (true no matter how the world might have been). For instance, logical truths are not just true because of some accidental feature of the world, but true come what may. A possible statement may not actually be true, but it might have been true. We use $\Box$ to express necessity, and $\Diamond$ to express possibility. So you can read $\Box X$ as *It is necessarily the case that* X, and $\Diamond X$ as *It is possibly the case that* X.

There are lots of different kinds of necessity. It is *humanly impossible* for me to run at 100mph. Given the sorts of creatures that we are, no human can do that. But still, it isn't *physically impossible* for me to run that fast. We haven't got the technology to do it yet, but it is surely physically possible to swap my biological legs for robotic ones which could run at 100mph. By contrast, it is physically impossible for me to run faster than the speed of light. The laws of physics forbid any object from accelerating up to that speed. But even that isn't *logically* impossible. It isn't a contradiction to imagine that the laws of physics might have been different, and that they might have allowed objects to move faster than light.

Which kind of modality does ML deal with? *All of them!* ML is a very flexible tool. We start with a basic set of rules that govern $\Box$ and $\Diamond$, and then add more rules to fit whatever kind of modality we are interested in. In fact, ML is so flexible that we do not even have to think of $\Box$ and $\Diamond$ as expressing *necessity* and *possibility*. We might instead read $\Box$ as expressing *provability*, so that $\Box X$ means *It is provable that X*, and $\Diamond X$ means *It is not refutable that X*. Similarly, we can interpret $\Box$ to mean *S knows that X* or *S believes that X*. Or we might read $\Box$ as expressing *moral obligation*, so that $\Box X$ means *It is morally obligatory that X*, and $\Diamond X$ means *It is morally permissible that X*. All we would need to do is cook up the right rules for these different readings of $\Box$ and $\Diamond$.

A modal formula is one that includes modal operators such as $\Box$ and $\Diamond$. Depending on the interpretation we assign to $\Box$ and $\Diamond$, different modal formulas will be provable or valid. For instance, $\Box X \to X$ might say that "if X is necessary, it is true," if $\Box$ is interpreted as necessity. It might express "if X is known, then it is true," if $\Box$ expresses known truth. Under both these interpretations, $\Box X \to X$ is valid: All necessary propositions are true come what may, so are true in the actual world. And if a proposition is known to be true, it must be true (one can't know something that's false). However, when $\Box$ is interpreted as "it is believed that" or "it ought to be the case that," $\Box X \to X$ is not valid: We can believe false propositions. Not every proposition that ought to be true is in fact true, e.g., "Every murderer will be brought to justice." This *ought* to be true, but it isn't.

We will consider different kinds of systems of ML. They differ in the rules of proof allowed, and in the semantics we use to define our logical notions. The different systems we'll consider are called **K**, **T**, **S4**, and **S5**. **K** is the basic system; everything that is valid or provable in **K** is also provable in the others. But there are some things that **K** does not prove, such as the formula $\Box A \to A$ for sentence letter A. So **K** is not an appropriate modal logic for necessity and possibility (where $\Box X \to X$ should be provable). This is provable in the system **T**, so **T** is more appropriate when dealing with necessity and possibiliity, but less apropriate when dealing with belief or obligation, since then $\Box X \to X$ should *not* (always) be provable. The perhaps best system of ML for necessity and possibility, and in any case the most widely accepted, is the strongest of the systems we consider, **S5**.

45.1 The Language of ML

In order to do modal logic, we have to do two things. First, we want to learn how to prove things in ML. Second, we want to see how to construct interpretations for ML. But before we can do either of these things, we need to explain how to construct sentences in ML.

The language of ML is an extension of TFL. We could have started with FOL, which would have given us Quantified Modal Logic (QML). QML is much more powerful than ML, but it is also much, much more complicated. So we are going to keep things simple, and start with TFL.

Just like TFL, ML starts with an infinite stock of *atoms*. These are written as capital letters, with or without numerical subscripts: A, B, $\ldots A_1$, B_1, $\ldots$ We then take all of the rules about how to make sentences from TFL, and add two more for $\Box$ and $\Diamond$:

(1) Every atom of ML is a sentence of ML.

(2) If X is a sentence of ML, then $\neg X$ is a sentence of ML.

(3) If X and Y are sentences of ML, then $(X \,\&\, Y)$ is a sentence of ML.

(4) If X and Y are sentences of ML, then $(X \vee Y)$ is a sentence of ML.

(5) If X and Y are sentences of ML, then $(X \rightarrow Y)$ is a sentence of ML.

(6) If X and Y are sentences of ML, then $(X \leftrightarrow Y)$ is a sentence of ML.

(7) If X is a sentence of ML, then $\Box X$ is a sentence of ML.

(8) If X is a sentence of ML, then $\Diamond X$ is a sentence of ML.

(9) Nothing else is a sentence of ML.

Here are some examples of ML sentences:

$$A,\ P \vee Q,\ \Box A,\ C \vee \Box D,\ \Box\Box(A \rightarrow R),\ \Box\Diamond(S \,\&\, (Z \leftrightarrow (\Box W \vee \Diamond Q)))$$

CHAPTER 46

Natural deduction for ML

Now that we know how to make sentences in ML, we can look at how to *prove* things in ML. We will use $\vdash$ to express provability. So $X_1, X_2, \dots X_n \vdash Z$ means that Z can be proven from $X_1, X_2, \dots X_n$. However, we will be looking at a number of different systems of ML, and so it will be useful to add a subscript to indicate which system we are working with. So for example, if we want to say that we can prove Z from $X_1, X_2, \dots X_n$ *in system* **K**, we will write: $X_1, X_2, \dots X_n \vdash_{\mathbf{K}} Z$.

46.1 System K

We start with a particularly simple system called **K**, in honour of the philosopher and logician Saul Kripke. **K** includes all of the natural deduction rules from TFL, including the derived rules as well as the basic ones. **K** then adds a special kind of subproof, plus two new basic rules for $\Box$.

The special kind of subproof looks like an ordinary subproof, except it has a $\Box$in its assumption line instead of a formula. We call them *strict subproofs*—they allow as to reason and prove things about alternate possibilities. What we can prove inside a strict subproof holds in any alternate possibility, in particular, in alternate possibilities where the assumptions in force in our proof may not hold. In a strict subproofs, all assumptions are disregarded, and we are not allowed to appeal to any lines outside the strict subproof (except as allowed by the modal rules given below).

The $\Box$I rule allows us to derive a formula $\Box X$ if we can derive X inside a strict subproof. It is our fundamental method of introducing $\Box$ into proofs. The basic idea is simple enough: if X is a theorem, then $\Box X$ should be a theorem too. (Remember that to call X a theorem is to say that we can prove X without relying on any undischarged assumptions.)

Suppose we wanted to prove $\Box(A \to A)$. The first thing we need to do is prove that $A \to A$ is a theorem. You already know how to do that using TFL. You simply present a proof of $A \to A$ which doesn't start with any premises, like this:

$$
\begin{array}{l|l l}
1 & \quad | \; \underline{A} & \\
2 & \quad | \; A & \text{R } 1 \\
3 & A \to A & \to\text{I } 1\text{–}2
\end{array}
$$

But to apply $\Box$I, we need to have proven the formula inside a strict subproof. Since our proof of $A \to A$ makes use of no assumptions at all, this is possible.

$$
\begin{array}{l|l l}
1 & \quad | \; \underline{\Box} & \\
2 & \quad | \quad | \; \underline{A} & \\
3 & \quad | \quad | \; A & \text{R } 2 \\
4 & \quad | \; A \to A & \to\text{I } 2\text{–}3 \\
5 & \Box(A \to A) & \Box\text{I } 1\text{–}4
\end{array}
$$

$$
\begin{array}{l|l l}
m & \quad | \; \underline{\Box} & \\
n & \quad | \; X & \\
 & \Box X & \Box\text{I } m\text{–}n
\end{array}
$$

No line above line m may be cited by any rule within the strict subproof begun at line m unless the rule explicitly allows it.

It is essential to emphasise that in strict subproof you cannot use any rule which appeals to anything you proved outside of the strict subproof. There are exceptions, e.g., the $\Box$E rule below. These rules will explicitly state that they can be used inside strict subproofs and cite lines outside the strict subproof. This restriction is essential, otherwise we would get terrible results. For example, we could provide the following proof to vindicate $A \therefore \Box A$:

1	A	
2	$\quad \Box$	
3	$\quad A$	incorrect use of R 1
4	$\Box A$	$\Box$I 2–3

This is not a legitimate proof, because at line 3 we appealed to line 1, even though line 1 comes before the beginning of the strict subproof at line 2.

We said above that a strict subproof allows us to reason about arbitrary alternate possible situations. What can be proved in a strict subproof holds in all alternate possible situtations, and so is necessary. This is the idea behind the $\Box$I rule. On the other hand, if we've assumed that something is necessary, we have therewith assumed that it is true in all alternate possbile situations. Hence, we have the rule $\Box$E:

m	$\Box X$	
	$\quad \Box$	
n	$\quad X$	$\Box$E m

$\Box$E can only be applied if line m (containing $\Box A$) lies *outside* of the strict subproof in which line n falls, and this strict subproof is not itself part of a strict subproof not containing m.

$\Box$E allows you to assert X inside a strict subproof if you have $\Box X$ outside the strict subproof. The restriction means that you can only do this in the first strict subproof, you cannot apply the $\Box$E rule inside a nested strict subproof. So the following is not allowed:

1	$\Box X$	
2	$\quad\vert\ \Box$	
3	$\quad\vert\quad\vert\ \Box$	
4	$\quad\vert\quad\vert\ X$	incorrect use of $\Box$E 1

The incorrect use of $\Box$E on line 4 violates the condition, because although line 1 lies outside the strict subproof in which line 4 falls, the strict subproof containing line 4 lies inside the strict subproof beginning on line 2 which does not contain line 1.

Let's begin with an example.

1	$\Box A$	
2	$\Box B$	
3	$\quad\vert\ \Box$	
4	$\quad\vert\ A$	$\Box$E 1
5	$\quad\vert\ B$	$\Box$E 2
6	$\quad\vert\ A \mathbin{\&} B$	&I 4, 5
7	$\Box(A \wedge B)$	$\Box$I 3–7

We can also mix regular subproofs and strict subproofs:

1	$\Box(A \to B)$	
2	$\quad\vert\ \Box A$	
3	$\quad\vert\quad\vert\ \Box$	
4	$\quad\vert\quad\vert\ A$	$\Box$E m
5	$\quad\vert\quad\vert\ A \to B$	$\Box$E 1
6	$\quad\vert\quad\vert\ B$	$\to$E 4, 5
7	$\quad\vert\ \Box B$	
8	$\Box A \to \Box B$	$\to$I 2–7

This is called the *Distribution Rule*, because it tells us that $\Box$ 'distributes' over $\rightarrow$.

The rules $\Box$I and $\Box$E look simple enough, and indeed **K** is a very simple system! But **K** is more powerful than you might have thought. You can prove a fair few things in it.

46.2 Possibility

In the last subsection, we looked at all of the basic rules for **K**. But you might have noticed that all of these rules were about necessity, $\Box$, and none of them were about possibility, $\Diamond$. That's because we can *define* possibility in terms of necessity:

$$\Diamond X =_{df} \neg\Box\neg X$$

In other words, to say that X is *possibly true*, is to say that X is *not necessarily false*. As a result, it isn't really essential to add a $\Diamond$, a special symbol for possibility, into system **K**. Still, the system will be *much* easier to use if we do, and so we will add the following definitional rules:

m	$\neg\Box\neg X$	
	$\Diamond X$	Def$\Diamond$ m

m	$\Diamond X$	
	$\neg\Box\neg X$	Def$\Diamond$ m

Importantly, you should not think of these rules as any real addition to **K**: they just record the way that $\Diamond$ is defined in terms of $\Box$.

If we wanted, we could leave our rules for **K** here. But it will be helpful to add some *Modal Conversion* rules, which give us some more ways of flipping between $\Box$ and $\Diamond$:

$$\begin{array}{r|ll} m & \neg\Box X & \\ & \Diamond\neg X & \text{MC } m \end{array}$$

$$\begin{array}{r|ll} m & \Diamond\neg X & \\ & \neg\Box X & \text{MC } m \end{array}$$

$$\begin{array}{r|ll} m & \neg\Diamond X & \\ & \Box\neg X & \text{MC } m \end{array}$$

$$\begin{array}{r|ll} m & \Box\neg X & \\ & \neg\Diamond X & \text{MC } m \end{array}$$

These Modal Conversion Rules are also no addition to the power of **K**, because they can be derived from the basic rules, along with the definition of $\Diamond$.

In system **K**, using Def$\Diamond$ (or the modal conversion rules), one can prove $\Diamond A \leftrightarrow \neg\Box\neg A$. When laying out system **K**, we started with $\Box$ as our primitive modal symbol, and then defined $\Diamond$ in terms of it. But if we had preferred, we could have started with $\Diamond$ as our primitive, and then defined $\Box$ as follows: $\Box X =_{df} \neg\Diamond\neg X$. There is, then, no sense in which necessity is somehow more *fundamental* than possibility. Necessity and possibility are exactly as fundamental as each other.

46.3 System T

So far we have focussed on **K**, which is a very simple modal system. **K** is so weak that it will not even let you prove X from $\Box X$. But if we are thinking of $\Box$ as expressing *necessity*, then we will want to be able to make this inference: if X is *necessarily true*, then it must surely be *true*!

This leads us to a new system, **T**, which we get by adding the following rule to **K**:

$$\begin{array}{r|l@{\qquad\qquad}l} m & \Box X & \\ n & X & \textbf{RT}\ m \end{array}$$

The line n on which rule **RT** is applied must *not* lie in a strict subproof that begins after line m.

The restriction on rule **T** is in a way the opposite of the restriction on □E: you can *only* use □E in a nested strict subproof, but you cannot use **T** in a nested strict subproof.

We can prove things in **T** which we could not prove in **K**, e.g., $\Box A \to A$.

46.4 System S4

T allows you to strip away the necessity boxes: from $\Box X$, you may infer X. But what if we wanted to add extra boxes? That is, what if we wanted to go from $\Box X$ to $\Box\Box X$? Well, that would be no problem, if we had proved $\Box X$ by applying □I to a strict subproof of X which itself does not use □E. In that case, X is a tautology, and by nesting the strict subproof inside another strict subproof and applying □I again, we can prove $\Box\Box X$. For example, we could prove $\Box\Box(P \to P)$ like this:

1	$\Box$	
2	$\Box$	
3	P	
4	P	R 3
5	$P \to P$	→I 3–4
6	$\Box(P \to P)$	□I 2–5
7	$\Box\Box(P \to P)$	□I 1–6

But what if we didn't prove $\Box X$ in this restricted way, but used □E inside the strict subproof of X. If we put that strict subproof inside another strict subproof, the requirement of rule □E to not cite a line containing $\Box X$ which lies in another strict subproof that has not yet concluded, is violated. Or what if $\Box X$ were just an assumption we started our proof with? Could we infer $\Box\Box X$ then? Not in **T**, we couldn't. And this might well strike you as a limitation of **T**, at least if we are reading $\Box$ as expressing *necessity*. It seems intuitive that if X is necessarily true, then it couldn't have *failed* to be necessarily true.

This leads us to another new system, **S4**, which we get by adding the following rule to **T**:

$$
\begin{array}{r|lll}
m & \Box X & & \\
 & & \multicolumn{1}{|l}{\Box} & \\ \cline{3-3}
n & & \multicolumn{1}{|l}{\Box X} & \text{R}\mathbf{4}\ m
\end{array}
$$

Note that R**4** can only be applied if line m (containing $\Box A$) lies outside of the strict subproof in which line n falls, and this strict subproof is not itself part of a strict subproof not containing n.

Rule R**4** looks just like $\Box$E, except that instead of yielding X from $\Box X$ it yields $\Box X$ inside a strict subproof. The restriction is the same, however: R**4** allows us to "import" $\Box X$ into a strict subproof, but not into a strict subproof itself nested inside a strict subproof. However, if that is necessary, an additional application of R**4** would have the same result.

Now we can prove even more results. For instance:

$$
\begin{array}{r|llll}
1 & & \multicolumn{1}{|l}{\Box A} & & \\ \cline{3-3}
2 & & \multicolumn{1}{|l}{} & \multicolumn{1}{|l}{\Box} & \\ \cline{4-4}
3 & & \multicolumn{1}{|l}{} & \multicolumn{1}{|l}{\Box A} & \text{R}\mathbf{4}\ 1 \\
4 & & \multicolumn{1}{|l}{\Box\Box A} & & \Box\text{I } 2\text{–}3 \\
5 & \Box A \to \Box\Box A & & & \to\text{I } 1\text{–}6
\end{array}
$$

Similarly, we can prove $\Diamond\Diamond A \to \Diamond A$. This shows us that as well as letting us *add* extra *boxes*, **S4** lets us *delete* extra *diamonds*: from $\Diamond\Diamond X$, you can always infer $\Diamond X$.

46.5 System S5

In **S4**, we can always add a box in front of another box. But **S4** does not automatically let us add a box in front of a *diamond*. That is, **S4** does not generally permit the inference from $\Diamond X$ to $\Box\Diamond X$. But again, that might strike you as a shortcoming, at least if you are reading $\Box$ and $\Diamond$ as expressing *necessity* and *possibility*. It seems intuitive that if X is possibly true, then it couldn't have *failed* to be possibly true.

This leads us to our final modal system, **S5**, which we get by adding the following rule to **S4**:

m	$\neg\Box X$	
	$\quad \Box$	
n	$\quad \neg\Box X$	**R5** m

Rule **R5** can only be applied if line m (containing $\neg\Box X$) lies outside of the strict subproof in which line n falls, and this strict subproof is not itself part of a strict subproof not containing line m.

This rule allows us to show, for instance, that $\Diamond\Box A \vdash_{\mathbf{S5}} \Box A$:

1	$\Diamond\Box A$	
2	$\neg\Box\neg\Box A$	Def$\Diamond$ 1
3	$\quad \neg\Box A$	
4	$\quad\quad \Box$	
5	$\quad\quad \neg\Box A$	**R5** 3
6	$\quad \Box\neg\Box A$	$\Box$I 4–5
7	$\quad \bot$	$\neg$E 2, 6
8	$\Box A$	IP 3–7

So, as well as adding boxes in front of diamonds, we can also delete diamonds in front of boxes.

We got **S5** just by adding the rule R**5** rule to **S4**. In fact, we could have added rule R**5** to **T** alone, and leave out rule R**4**). Everything we can prove by rule R**4** can also be proved using R**T** together with R**5**. For instance, here is a proof that shows $\Box A \vdash_{\mathbf{S5}} \Box\Box A$ without using R**4**:

1	$\mid \Box A$	
2	$\mid \quad \mid \Box\neg\Box A$	
3	$\mid \quad \mid \neg\Box A$	R**T** 2
4	$\mid \quad \mid \bot$	$\neg$E 1, 3
5	$\mid \neg\Box\neg\Box A$	$\neg$I 2–4
6	$\mid \quad \mid \Box$	
7	$\mid \quad \mid \quad \mid \neg\Box A$	
8	$\mid \quad \mid \quad \mid \quad \mid \Box$	
9	$\mid \quad \mid \quad \mid \quad \mid \neg\Box A$	R**5** 7
10	$\mid \quad \mid \quad \mid \Box\neg\Box A$	$\Box$I 8–9
11	$\mid \quad \mid \quad \mid \neg\Box\neg\Box A$	R**5** 5
12	$\mid \quad \mid \quad \mid \bot$	$\neg$E 10, 11
13	$\mid \quad \mid \Box A$	PbC 7–12
14	$\mid \Box\Box A$	$\Box$I 6–13

S5 is *strictly stronger* than **S4**: there are things which can be proved in **S5**, but not in **S4** (e.g., $\Diamond\Box A \to \Box A$).

The important point about **S5** can be put like this: if you have a long string of boxes and diamonds, in any combination whatsoever, you can delete all but the last of them. So for example, $\Diamond\Box\Diamond\Diamond\Box\Box\Diamond\Box A$ can be simplified down to just $\Box A$.

CHAPTER 47

Semantics for ML

So far, we have focussed on laying out various systems of Natural Deduction for ML. Now we will look at the *semantics* for ML. A semantics for a language is a method for assigning truth-values to the sentences in that language. So a semantics for ML is a method for assigning truth-values to the sentences of ML.

47.1 Interpretations of ML

The big idea behind the semantics for ML is this. In ML, sentences are not just true or false, full stop. A sentence is true or false *at a given possible world*, and a single sentence may well be true at some worlds and false at others. We then say that $\Box X$ is true iff X is true at *every* world, and $\Diamond X$ is true iff X is true at *some* world.

That's the big idea, but we need to refine it and make it more precise. To do this, we need to introduce the idea of an *interpretation* of ML. The first thing you need to include in an interpretation is a collection of *possible worlds*. Now, at this point you might well want to ask: What exactly is a possible world? The intuitive idea is that a possible world is another way that this world could have been. But what exactly does that mean? This is an excellent philosophical question, and we will look at it in a lot of detail later. But we do not need to worry too much about it right now. As far as the formal logic goes, possible worlds can be anything you like. All that matters is that you supply each interpretation with a non-empty collection of things labelled POSSIBLE WORLDS.

Once you have chosen your collection of possible worlds, you need to find some way of determining which sentences of ML are true at which possible worlds. To do that, we need to introduce the notion of a *valuation function*. Those of you who have studied some maths will already be familiar with the general idea of a function. But for those of you who haven't, a function is a mathematical entity which maps arguments to values. That might sound a little bit abstract, but some familiar examples will help. Take the function $x + 1$. This is a function which takes in a number as argument, and then spits out the next number as value. So if you feed in the number 1 as an argument, the function $x+1$ will spit out the number 2 as a value; if you feed in 2, it will spit out 3; if you feed in 3, it will spit out 4 ... Or here is another example: the function $x+y$. This time, you have to feed two arguments into this function if you want it to return a value: if you feed in 2 and 3 as your arguments, it spits out 5; if you feed in 1003 and 2005, it spits out 3008; and so on.

A valuation function for ML takes in a sentence and a world as its arguments, and then returns a truth-value as its value. So if ν is a valuation function and w is a possible world, $\nu_w(X)$ is whatever truth-value ν maps X and w to: if $\nu_w(X) = F$, then X is false at world w on valuation ν; if $\nu_w(X) = T$, then X is true at world w on valuation ν.

These valuation functions are allowed to map any *atomic* sentence to any truth-value at any world. But there are rules about which truth-values more complex sentences get assigned at a world. Here are the rules for the connectives from TFL:

(1) $\nu_w(\neg X) = T$ iff: $\nu_w(X) = F$

(2) $\nu_w(X \,\&\, Y) = T$ iff: $\nu_w(X) = T$ and $\nu_w(Y) = T$

(3) $\nu_w(X \vee Y) = T$ iff: $\nu_w(X) = T$ or $\nu_w(Y) = T$, or both

(4) $\nu_w(X \rightarrow Y) = T$ iff: $\nu_w(X) = F$ or $\nu_w(Y) = T$, or both

(5) $\nu_w(X \leftrightarrow Y) = T$ iff: $\nu_w(X) = T$ and $\nu_w(Y) = T$, or $\nu_w(X) = F$ and $\nu_w(Y) = F$

So far, these rules should all look very familiar. Essentially, they all work exactly like the truth-tables for TFL. The only difference is that these truth-table rules have to be applied over and over again, to one world at a time.

But what are the rules for the new modal operators, $\Box$ and $\Diamond$? The most obvious idea would be to give rules like these:

$$\nu_w(\Box X) = T \text{ iff } \forall w'(\nu_{w'}(X) = T)$$

$$\nu_w(\Diamond X) = T \text{ iff } \exists w'(\nu_{w'}(X) = T)$$

This is just the fancy formal way of writing out the idea that $\Box X$ is true at w just in case X is true at *every* world, and $\Diamond X$ is true at w just in case X is true at *some* world.

However, while these rules are nice and simple, they turn out not to be quite as useful as we would like. As we mentioned, ML is meant to be a very flexible tool. It is meant to be a general framework for dealing with lots of different kinds of necessity. As a result, we want our semantic rules for $\Box$ and $\Diamond$ to be a bit less rigid. We can do this by introducing another new idea: *accessibility relations.*

An accessibility relation, R, is a relation between possible worlds. Roughly, to say that Rw_1w_2 (in English: world w_1 *accesses* world w_2) is to say that w_2 is possible *relative to* w_1. In other words, by introducing accessibility relations, we open up the idea that a given world might be possible relative to some worlds but not others. This turns out to be a *very* fruitful idea when studying modal systems. We can now give the following semantic rules for $\Box$ and $\Diamond$:

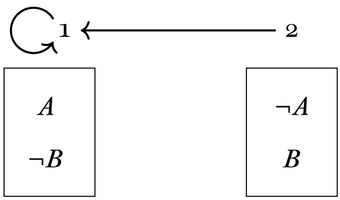
(6) $\nu_{w_1}(\Box X) = T \text{ iff } \forall w_2(Rw_1w_2 \rightarrow \nu_{w_2}(X) = T)$

(7) $\nu_{w_1}(\Diamond X) = T \text{ iff } \exists w_2(Rw_1w_2 \ \& \ \nu_{w_2}(X) = T)$

Or in plain English: $\Box X$ is true in world w_1 iff X is true in every world that is possible relative to w_1; and $\Diamond X$ is true in world w_1 iff X is true in some world that is possible relative to w_1.

So, there we have it. An interpretation for ML consists of three things: a collection of possible worlds, W; an accessibility relation, R; and a valuation function, ν. The collection of 'possible worlds' can really be a collection of anything you like. It really doesn't matter, so long as W isn't empty. (For many purposes, it is helpful just to take a collection of numbers to be your collection of worlds.) And for now, at least, R can be any relation between the worlds in W that you like. It could be a relation which every world in W bears to every world in W, or one which no world bears to any world, or anything in between. And lastly, ν can map any atomic sentence of ML to any truth-value at any world. All that matters is that it follows the rules (1)–(7) when it comes to the more complex sentences.

Let's look at an example. It is often helpful to present interpretations of ML as diagrams, like this:



Here is how to read the interpretation off from this diagram. It contains just two worlds, 1 and 2. The arrows between the worlds indicate the accessibility relation. So 1 and 2 both access 1, but neither 1 nor 2 accesses 2. The boxes at each world let us know which atomic sentences are true at each world: A is true at 1 but false at 2; B is false at 1 but true at 2. You may only write an atomic sentence or the negation of an atomic sentence into one of these boxes. We can figure out what truth-values the more complex sentences get at each world from that. For example, on this interpretation all of the following sentences are true at w_1:

$$A \mathbin{\&} \neg B,\ B \rightarrow A,\ \Diamond A,\ \Box\neg B$$

If you don't like thinking diagrammatically, then you can also present an interpretation like this:

W: $1, 2$

R: $\langle 1,1\rangle, \langle 2,1\rangle$

$$\nu_1(A) = T, \nu_2(B) = F, \nu_2(A) = F, \nu_2(B) = T$$

You will get the chance to cook up some interpretations of your own shortly, when we start looking at *counter-interpretations*.

47.2 A Semantics for System K

We can now extend all of the semantic concepts of TFL to cover ML:

- ▹ $X_1, X_2, \ldots X_n \therefore Z$ is modally valid iff there is no world in any interpretation at which $X_1, X_2, \ldots X_n$ are all true and Z is false.

- ▹ X is a modal truth iff X is true at every world in every interpretation.

- ▹ X is a modal contradiction iff X is false at every world in every interpretation.

- ▹ X is modally satisfiable iff X is true at some world in some interpretation.

(From now on we will drop the explicit 'modal' qualifications, since they can be taken as read.)

We can also extend our use of $\vDash$. However, we need to add subscripts again, just as we did with $\vdash$. So, when we want to say that $X_1, X_2, \ldots X_n \therefore Z$ is valid, we will write: $X_1, X_2, \ldots X_n \vDash_{\mathbf{K}} Z$.

Let's get more of a feel for this semantics by presenting some counter-interpretations. Consider the following (false) claim:

$$\neg A \vDash_{\mathbf{K}} \neg\Diamond A$$

In order to present a counter-interpretation to this claim, we need to cook up an interpretation which makes $\neg A$ true at some world w, and $\neg\Diamond A$ false at w. Here is one such interpretation, presented diagrammatically:

It is easy to see that this will work as a counter-interpretation for our claim. First, $\neg A$ is true at world 1. And second, $\neg\Diamond A$ is false at 1: A is true at 2, and 2 is accessible from 1. So there is some world in this interpretation where $\neg A$ is true and $\neg\Diamond A$ is false, so it is not the case that $\neg A \vDash_{\mathbf{K}} \neg\Diamond A$.

Why did we choose the subscript **K**? Well, it turns out that there is an important relationship between system **K** and the definition of validity we have just given. In particular, we have the following two results:

- ▹ If $X_1, X_2, \ldots X_n \vdash_{\mathbf{K}} Z$, then $X_1, X_2, \ldots X_n \vDash_{\mathbf{K}} Z$
- ▹ If $X_1, X_2, \ldots X_n \vDash_{\mathbf{K}} Z$, then $X_1, X_2, \ldots X_n \vdash_{\mathbf{K}} Z$

The first result is known as a *soundness* result, since it tells us that the rules of **K** are good, sound rules: if you can vindicate an argument by giving a proof for it using system **K**, then that argument really is valid. The second result is known as a *completeness* result, since it tells us that the rules of **K** are broad enough to capture all of the valid arguments: if an argument is valid, then it will be possible to offer a proof in **K** which vindicates it.

Now, it is one thing to state these results, quite another to prove them. However, we will not try to prove them here. But the idea behind the proof of soundness will perhaps make clearer how strict subproofs work.

In a strict subproof, we are not allowed to make use of any information from outside the strict subproof, except what we import into the strict subproof using □E. If we've assumed or proved $\Box X$, by □E, we can used X inside a strict subproof. And in **K**, that is the only way to import a formula into a strict subproof. So everything that can be proved inside a strict subproof must follow from formulas X where outside the strict subproof we have $\Box X$. Let's imagine that we are reasoning about what's true in a possible world in some interpretation. If we know that $\Box X$ is true in that possible world, we know that X is true in all accessible worlds. So, everything proved inside a strict subproof is true in all accessible possible worlds. That is why □I is a sound rule.

47.3 A Semantics for System T

A few moments ago, we said that system **K** is sound and complete. Where does that leave the other modal systems we looked at, namely **T**, **S4** and **S5**? Well, they are all *unsound*, relative to the definition of validity we gave above. For example, all of these systems allow us to infer A from $\Box A$, even though $\Box A \nvDash_{\mathbf{K}} A$.

Does that mean that these systems are a waste of time? Not at all! These systems are only unsound *relative to the definition of validity we gave above.* (Or to use symbols, they are unsound relative to $\vDash_{\mathbf{K}}$.) So when we are dealing with these stronger modal systems, we just need to modify our definition of validity to fit. This is where accessibility relations come in really handy.

When we introduced the idea of an accessibility relation, we said that it could be any relation between worlds that you like: you could have it relating every world to every world, no world to any world, or anything in between. That is how we were thinking of accessibility relations in our definition of $\vDash_{\mathbf{K}}$. But if we wanted, we could start putting some restrictions on the accessibility relation. In particular, we might insist that it has to be *reflexive*:

▹ $\forall w R w w$

In English: every world accesses itself. Or in terms of relative possibility: every world is possible relative to itself. If we imposed this restriction, we could introduce a new consequence relation, $\vDash_{\mathbf{T}}$, as follows:

> $X_1, X_2, \ldots X_n \vDash_{\mathbf{T}} Z$ iff there is no world in any interpretation *which has a reflexive accessibility relation*, at which $X_1, X_2, \ldots X_n$ are all true and Z is false

We have attached the **T** subscript to $\vDash$ because it turns out that system **T** is sound and complete relative to this new definition of validity:

▹ If $X_1, X_2, \ldots X_n \vdash_{\mathbf{T}} Z$, then $X_1, X_2, \ldots X_n \vDash_{\mathbf{T}} Z$

▹ If $X_1, X_2, \ldots X_n \vDash_{\mathbf{T}} Z$, then $X_1, X_2, \ldots X_n \vdash_{\mathbf{T}} Z$

As before, we will not try to prove these soundness and completeness results. However, it is relatively easy to see how insisting that the accessibility relation must be reflexive will vindicate the R**T** rule:

m	$\Box X$	
	X	R**T** m

To see this, just imagine trying to cook up a counter-interpretation to this claim:

$$\Box X \vDash_{\mathbf{T}} X$$

We would need to construct a world, w, at which $\Box X$ was true, but X was false. Now, if $\Box X$ is true at w, then X must be true at every world w accesses. But since the accessibility relation is reflexive, w accesses w. So X must be true at w. But now X must be true *and* false at w. Contradiction!

47.4 A Semantics for S4

How else might we tweak our definition of validity? Well, we might also stipulate that the accessibility relation has to be *transitive*:

- ▹ $\forall w_1 \forall w_2 \forall w_3((Rw_1w_2 \,\&\, Rw_2w_3) \to Rw_1w_3)$

In English: if w_1 accesses w_2, and w_2 accesses w_3, then w_1 accesses w_3. Or in terms of relative possibility: if w_3 is possible relative to w_2, and w_2 is possible relative to w_1, then w_3 is possible relative to w_1. If we added this restriction on our accessibility relation, we could introduce a new consequence relation, $\vDash_{\mathbf{S4}}$, as follows:

> $X_1, X_2, \ldots X_n \vDash_{\mathbf{S4}} Z$ iff there is no world in any interpretation *which has a reflexive and transitive accessibility relation*, at which $X_1, X_2, \ldots X_n$ are all true and Z is false

We have attached the **S4** subscript to $\vDash$ because it turns out that system **S4** is sound and complete relative to this new definition of validity:

- ▹ If $X_1, X_2, \ldots X_n \vdash_{\mathbf{S4}} Z$, then $X_1, X_2, \ldots X_n \vDash_{\mathbf{S4}} Z$
- ▹ If $X_1, X_2, \ldots X_n \vDash_{\mathbf{S4}} Z$, then $X_1, X_2, \ldots X_n \vdash_{\mathbf{S4}} Z$

As before, we will not try to prove these soundness and completeness results. However, it is relatively easy to see how insisting that the accessibility relation must be transitive will vindicate the **S4** rule:

$$\begin{array}{r|l}
m & \Box X \\
 & \quad \begin{array}{|l} \Box \\ \hline \Box X \qquad \text{R4 } m \end{array}
\end{array}$$

The idea behind strict subproofs, remember, is that they are ways to prove things that must be true in all accessible worlds. So the R4 rule means that whenever $\Box X$ is true, $\Box X$ must also be true in every accessible world. In other words, we must have $\Box X \vDash_{\mathbf{S4}} \Box\Box X$.

To see this, just imagine trying to cook up a counter-interpretation to this claim:

$\Box X \vDash_{\mathbf{S4}} \Box\Box X$

We would need to construct a world, w_1, at which $\Box X$ was true, but $\Box\Box X$ was false. Now, if $\Box\Box X$ is false at w_1, then w_1 must access some world, w_2, at which $\Box X$ is false. Equally, if $\Box X$ is false at w_2, then w_2 must access some world, w_3, at which X is false. We just said that w_1 accesses w_2, and w_2 accesses w_3. So since we are now insisting that the accessibility relation be transitive, w_1 must access w_3. And as $\Box X$ is true at w_1, and w_3 is accessible from w_1, it follows that X must be true at w_3. So X is true *and* false at w_3. Contradiction!

47.5 A Semantics for S5

Let's put one more restriction on the accessibility relation. This time, let's insist that it must also be *symmetric*:

- $\forall w_1 \forall w_2 (Rw_1w_2 \rightarrow Rw_2w_1)$

In English: if w_1 accesses w_2, then w_2 accesses w_1. Or in terms of relative possibility: if w_2 is possible relative to w_1, then w_1 is possible relative to w_2. Logicians call a relation that is reflexive, symmetric, and transitive an *equivalence* relation. We can now define a new consequence relation, $\vDash_{\mathbf{S5}}$, as follows:

> $X_1, X_2, \ldots X_n \vDash_{\mathbf{S5}} Z$ iff there is no world in any interpretation *whose accessibility relation is an equivalence relation*, at which $X_1, X_2, \ldots X_n$ are all true and Z is false

We have attached the **S5** subscript to $\vDash$ because it turns out that system **S5** is sound and complete relative to this new definition of validity:

- If $X_1, X_2, \ldots X_n \vdash_{\mathbf{S5}} Z$, then $X_1, X_2, \ldots X_n \vDash_{\mathbf{S5}} Z$
- If $X_1, X_2, \ldots X_n \vDash_{\mathbf{S5}} Z$, then $X_1, X_2, \ldots X_n \vdash_{\mathbf{S5}} Z$

As before, we will not try to prove these soundness and completeness results here. However, it is relatively easy to see how insisting that the accessibility relation must be an equivalence relation will vindicate the R5 rule:

$$
\begin{array}{r|l}
m & \neg\Box X \\
& \begin{array}{|l}
\Box \\
\hline
\neg\Box X \qquad \text{R5 } m
\end{array}
\end{array}
$$

The rule says that if X is not necessary, i.e., false in some accessible world, it is also not necessary in any accessible prossible world, i.e., we have $\neg\Box X \vdash_{\mathbf{S5}} \Box\neg\Box X$.

To see this, just imagine trying to cook up a counter-interpretation to this claim:

$$\neg\Box X \vDash_{\mathbf{S5}} \Box\neg\Box X$$

We would need to construct a world, w_1, at which $\neg\Box X$ was true, but $\Box\neg\Box X$ was false. Now, if $\neg\Box X$ is true at w_1, then w_1 must access some world, w_2, at which X is false. Equally, if $\Box\neg\Box X$ is false at w_1, then w_1 must access some world, w_3, at which $\neg\Box X$ is false. Since we are now insisting that the accessibility relation is an equivalence relation, and hence symmetric, we can infer that w_3 accesses w_1. Thus, w_3 accesses w_1, and w_1 accesses w_2. Again, since we are now insisting that the accessibility relation is an equivalence relation, and hence transitive, we can infer that w_3 accesses w_2. But earlier we said that $\neg\Box X$ is false at w_3, which implies that X is true at every world which w_3 accesses. So X is true *and* false at w_2. Contradiction!

In the definition of $\vDash_{\mathbf{S5}}$, we stipulated that the accessibility relation must be an equivalence relation. But it turns out that there is another way of getting a notion of validity fit for **S5**. Rather than stipulating that the accessibility relation be an equivalence relation, we can instead stipulate that it be a *universal* relation:

- $\forall w_1 \forall w_2 R w_1 w_2$

In English: every world accesses every world. Or in terms of relative possibility: every world is possible relative to every world. Using this restriction on the accessibility relation, we could have defined $\vDash_{\mathbf{S5}}$ like this:

$X_1, X_2, \ldots X_n \vDash_{\mathbf{S5}} Z$ iff there is no world in any interpretation *which has a universal accessibility relation*, at which $X_1, X_2, \ldots X_n$ are all true and Z is false.

If we defined $\vDash_{\mathbf{S5}}$ like this, we would still get the same soundness and completeness results for **S5**. What does this tell us? Well, it means that if we are dealing with a notion of necessity according to which *every* world is possible relative to *every* world, then we should use **S5**. What is more, most philosophers assume that the notions of necessity that they are most concerned with, like *logical necessity* and *metaphysical necessity*, are of exactly this kind. So **S5** is the modal system that most philosophers use most of the time.

Further reading

Modal logic is a large subfield of logic. We have only scratched the surface. If you want to learn more about modal logic, here are some textbooks you might consult.

- ▹ Hughes, G. E., & Cresswell, M. J. (1996). *A New Introduction to Modal Logic*, Oxford: Routledge.
- ▹ Priest, G. (2008). *An Introduction to Non-Classical Logic*, 2nd ed., Cambridge: Cambridge University Press.
- ▹ Garson, J. W. (2013). *Modal Logic for Philosophers*, 2nd ed., Cambridge: Cambridge University Press.

None of these authors formulate their modal proof systems in quite the way we did, but the closest formulation is given by Garson.

Appendices

APPENDIX A

Symbolic notation

1.1 Alternative nomenclature

Truth-functional logic. TFL goes by other names. Sometimes it is called *sentential logic*, because it deals fundamentally with sentences. Sometimes it is called *propositional logic*, on the idea that it deals fundamentally with propositions. We have stuck with *truth-functional logic*, to emphasize the fact that it deals only with assignments of truth and falsity to sentences, and that its connectives are all truth-functional.

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

Formulas. Some texts call formulas *well-formed formulas*. Since 'well-formed formula' is such a long and cumbersome phrase, they then abbreviate this as *wff*. This is both barbarous and unnecessary (such texts do not countenance 'ill-formed formulas'). We have stuck with 'formula'.

In §5, we defined *sentences* of TFL. These are also sometimes called 'formulas' (or 'well-formed formulas') since in TFL, unlike FOL, there is no distinction between a formula and a sentence.

Valuations. Some texts call valuations *truth-assignments*, or *truth-value assignments*.

Expressive adequacy. Some texts describe TFL as *truth-functionally complete*, rather than expressively adequate.

n-place predicates. We have chosen to call predicates 'one-place', 'two-place', 'three-place', etc. Other texts respectively call them 'monadic', 'dyadic', 'triadic', etc. Still other texts call them 'unary', 'binary', 'ternary', etc.

Names. In FOL, we have used 'a', 'b', 'c', for names. Some texts call these 'constants'. Other texts do not mark any difference between names and variables in the syntax. Those texts focus simply on whether the symbol occurs *bound* or *unbound*.

Domains. Some texts describe a domain as a 'domain of discourse', or a 'universe of discourse'.

1.2 Alternative symbols

In the history of formal logic, different symbols have been used at different times and by different authors. Often, authors were forced to use notation that their printers could typeset.

This appendix presents some common symbols, so that you can recognize them if you encounter them in an article or in another book.

Negation. Two commonly used symbols are the *hoe*, '$\neg$', and the *swung dash* or *tilda*, '$\sim$.' There are some issues typing '$\neg$' on a keyboard, and '$\sim$' is perfectly acceptable for you to use. In some more advanced formal systems it is necessary to distinguish between two kinds of negation; the distinction is sometimes represented by using both '$\neg$' and '$\sim$'. Older texts sometimes indicate negation by a line over the formula being negated, e.g., $\overline{A \,\&\, B}$. Some texts use '$x \neq y$' to abbreviate '$\neg\, x = y$'.

Disjunction. The symbol '$\vee$' is typically used to symbolize inclusive disjunction. One etymology is from the Latin word 'vel', meaning 'or'.

Conjunction. The two symbols commonly used for conjuction are *wedge*, '$\wedge$', and *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 '*&*'.) We have chosen to use it to allow for easier typing on a keyboard during these online-heavy times. However there are some substantial reservations about this choice. This symbol is commonly used in natural English writing (e.g. 'Smith & Sons'), and so even though it is a natural choice, many logicians use a different symbol to avoid confusion between the object and metalanguage: as a symbol in a formal system, the ampersand is not the English word '&'. The most common choice now is '$\wedge$', which is a counterpart to the symbol used for disjunction. Sometimes a single dot, '$\bullet$', is used. In some older texts, there is no symbol for conjunction at all; 'A and B' is simply written 'AB'.

Material Conditional. There are two common symbols for the material conditional: the *arrow*, '$\rightarrow$', and the *hook*, '$\supset$'.

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

Quantifiers. The universal quantifier is typically symbolized as a rotated 'A', and the existential quantifier as a rotated, 'E'. In some texts, there is no separate symbol for the universal quantifier. Instead, the variable is just written in parentheses in front of the formula that it binds. For example, they might write '$(x)Px$' where we would write '$\forall x\, Px$'.

These alternative typographies are summarised below:

negation	$\neg$, $\sim$
conjunction	$\wedge$, &, $\bullet$
disjunction	$\vee$
conditional	$\rightarrow$, $\supset$
biconditional	$\leftrightarrow$, $\equiv$
universal quantifier	$\forall x$, (x)

APPENDIX B

Alternative proof systems

In formulating our natural deduction system, we treated certain rules of natural deduction as *basic*, and others as *derived*. However, we could equally well have taken various different rules as basic or derived. We will illustrate this point by considering some alternative treatments of disjunction, negation, and the quantifiers. We will also explain why we have made the choices that we have.

2.1 Alternative disjunction elimination

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

$$
\begin{array}{l|ll}
m & X \vee Y & \\
i & \quad\vert\; \underline{X} & \\
 & \quad\vert\; \vdots & \\
j & \quad\vert\; C & \\
k & \quad\vert\; \underline{Y} & \\
 & \quad\vert\; \vdots & \\
l & \quad\vert\; C & \\
n & X \rightarrow C & {\rightarrow}\mathrm{I}\ i\text{–}j \\
n+1 & Y \rightarrow C & {\rightarrow}\mathrm{I}\ k\text{–}l \\
n+2 & C \vee \neg C & \mathrm{LEM} \\
n+3 & \quad\vert\; \underline{C} & \\
n+4 & \quad\vert\; C & \mathrm{R}\ n+3 \\
n+5 & \quad\vert\; \underline{\neg C} & \\
n+6 & \quad\vert\quad\vert\; \underline{X} & \\
n+7 & \quad\vert\quad\vert\; C & {\rightarrow}\mathrm{E}\ n,\ n+6 \\
n+8 & \quad\vert\quad\vert\; \bot & {\bot}\mathrm{I}\ n+7,\ n+5 \\
n+9 & \quad\vert\; \neg X & {\neg}\mathrm{I}\ n+6\text{–}n+8 \\
n+10 & \quad\vert\; Y & \mathrm{DS}\ m,\ n+9 \\
n+11 & \quad\vert\; C & {\rightarrow}\mathrm{E}\ n+1,\ n+10 \\
n+12 & C & {\vee}\mathrm{E}\ n+2,\ n+3\text{–}n+4,\ n+5\text{–}n+11
\end{array}
$$

So why did we choose to take ∨E as basic, rather than DS?[1] Our reasoning is that DS involves the use of '¬' in the statement of the rule. It is in some sense 'cleaner' for our disjunction elimination rule to avoid mentioning *other* connectives. The rule ∨E we use is also closely connected to the rule ∃E. Whereas there is no such analogy with DS.

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

2.2 Alternative negation rules

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

$$
\begin{array}{l|ll}
m & \quad \vert\ \underline{X} & \\
n-1 & \quad \vert\ Y & \\
n & \quad \vert\ \neg Y & \\
 & \neg X & \neg\mathrm{I}^{*}\ m\text{–}n
\end{array}
$$

and the following as their basic negation elimination rule:

$$
\begin{array}{l|ll}
m & \quad \vert\ \underline{\neg X} & \\
n-1 & \quad \vert\ Y & \\
n & \quad \vert\ \neg Y & \\
 & X & \neg\mathrm{E}^{*}\ m\text{–}n
\end{array}
$$

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

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

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

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

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

2.3 Alternative quantification rules

An alternative approach to the quantifiers is to take as basic the rules for ∀I and ∀E from §34, and also two CQ rule which allow us to move from $\forall x \neg X$ to $\neg \exists x X$ and vice versa.[3]

Taking only these rules as basic, we could have derived the ∃I and ∃E rules provided in §34. To derive the ∃I rule is fairly simple. Suppose X contains the name c, and contains no instances of the variable x, and that we want to do the following:

$$
\begin{array}{l|l}
m & X(\ldots c \ldots c \ldots) \\
k & \exists x X(\ldots x \ldots c \ldots)
\end{array}
$$

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

$$
\begin{array}{l|l|l}
m & X(\ldots c \ldots c \ldots) & \\
\hline
m+1 & \quad \neg\exists x X(\ldots x \ldots c \ldots) & \\
m+2 & \quad \forall x \neg X(\ldots x \ldots c \ldots) & \text{CQ } m+1 \\
m+3 & \quad \neg X(\ldots c \ldots c \ldots) & \forall\text{E } m+2 \\
m+4 & \quad \bot & \bot\text{I } m,\ m+3 \\
m+5 & \neg\neg\exists x X(\ldots x \ldots c \ldots) & \neg\text{I } m+1\text{–}m+4 \\
m+6 & \exists x X(\ldots x \ldots c \ldots) & \text{DNE } m+5
\end{array}
$$

To derive the ∃E rule is rather more subtle. This is because the ∃E rule has an important constraint (as, indeed, does the ∀I rule), and we need to make sure that we are respecting it. So, suppose we are in a situation where we *want* to do the following:

$$
\begin{array}{l|l}
m & \exists x X(\ldots x \ldots x \ldots) \\
i & \quad X(\ldots c \ldots c \ldots) \\
j & \quad Y \\
k & Y
\end{array}
$$

[3] Warren Goldfarb follows this line in *Deductive Logic*, 2003, Hackett Publishing Co.

where c does not occur in any undischarged assumptions, or in Y, or in $\exists x X(\ldots x \ldots x \ldots)$. Ordinarily, we would be allowed to use the ∃E rule; but we are not here assuming that we have access to this rule as a basic rule. Nevertheless, we could offer the following, more complicated derivation:

$$
\begin{array}{l|l|l}
m & \exists x X(\ldots x \ldots x \ldots) & \\
i & \quad \mid \underline{X(\ldots c \ldots c \ldots)} & \\
j & \quad \mid Y & \\
k & X(\ldots c \ldots c \ldots) \to Y & \to\mathrm{I}\ i\text{–}j \\
k+1 & \quad \mid \underline{\neg Y} & \\
k+2 & \quad \mid \neg X(\ldots c \ldots c \ldots) & \mathrm{MT}\ k,\ k+1 \\
k+3 & \quad \mid \forall x \neg X(\ldots x \ldots x \ldots) & \forall\mathrm{I}\ k+2 \\
k+4 & \quad \mid \neg \exists x X(\ldots x \ldots x \ldots) & \mathrm{CQ}\ k+3 \\
k+5 & \quad \mid \bot & \bot\mathrm{I}\ m,\ k+4 \\
k+6 & \neg\neg Y & \neg\mathrm{I}\ k+1\text{–}k+5 \\
k+7 & Y & \mathrm{DNE}\ k+6
\end{array}
$$

We are permitted to use ∀I on line $k+3$ because c does not occur in any undischarged assumptions or in Y. The entries on lines $k+4$ and $k+1$ contradict each other, because c does not occur in $\exists x X(\ldots x \ldots x \ldots)$.

Armed with these derived rules, we could now go on to derive the two remaining CQ rules, exactly as in §36.

So, why did we start with all of the quantifier rules as basic, and then derive the CQ rules?

Our first reason is that it seems more intuitive to treat the quantifiers as on a par with one another, giving them their own basic rules for introduction and elimination.

Our second reason relates to the discussion of alternative negation rules. In the derivations of the rules of ∃I and ∃E that we have offered in this section, we invoked DNE. This is a derived rule, whose derivation essentially depends upon the use of LEM. But, as we mentioned earlier, LEM is a contentious rule. So, if we want to move to a system which abandons LEM, but which still allows us to use existential quantifiers, we will want to take the introduction and elimination rules for the quantifiers as basic, and take the CQ rules as derived. (Indeed, in a system without LEM, we will be *unable* to derive the CQ rule which moves from $\neg\forall x X$ to $\exists x\neg X$.)

APPENDIX C

Quick reference

3.1 Sentences of TFL

Definition of being a sentence of TFL:

1. Every atomic sentence is a sentence.

    ▹ $A, B, C \ldots, W$, or with subscripts $A_1, B_3, A_{100}, J_{375}$

2. If X is a sentence, then $\neg X$ is a sentence.
3. If X and Y are sentences, then $(X \mathbin{\&} Y)$ is a sentence.
4. If X and Y are sentences, then $(X \vee Y)$ is a sentence.
5. If X and Y are sentences, then $(X \rightarrow Y)$ is a sentence.
6. If X and Y are sentences, then $(X \leftrightarrow Y)$ is a sentence.
7. Nothing else is a sentence.

3.2 Truth Rules for Connectives in TFL

$$\neg: \begin{array}{l} \text{T} \rightsquigarrow \text{F} \\ \text{F} \rightsquigarrow \text{T} \end{array} \qquad \&: \begin{array}{l} \text{T, T} \rightsquigarrow \text{T} \\ \text{T, F} \rightsquigarrow \text{F} \\ \text{F, T} \rightsquigarrow \text{F} \\ \text{F, F} \rightsquigarrow \text{F} \end{array} \qquad \vee: \begin{array}{l} \text{T, T} \rightsquigarrow \text{T} \\ \text{T, F} \rightsquigarrow \text{T} \\ \text{F, T} \rightsquigarrow \text{T} \\ \text{F, F} \rightsquigarrow \text{F} \end{array}$$

$$\rightarrow: \begin{array}{l} \text{T, T} \rightsquigarrow \text{T} \\ \text{T, F} \rightsquigarrow \text{F} \\ \text{F, T} \rightsquigarrow \text{T} \\ \text{F, F} \rightsquigarrow \text{T} \end{array} \qquad \leftrightarrow: \begin{array}{l} \text{T, T} \rightsquigarrow \text{T} \\ \text{T, F} \rightsquigarrow \text{F} \\ \text{F, T} \rightsquigarrow \text{F} \\ \text{F, F} \rightsquigarrow \text{T} \end{array}$$

3.3 Symbolization

Rough Meaning of the TFL Connectives

symbol	name	rough meaning
$\neg$	negation	'It is not the case that...'
&	conjunction	'... and ...'
$\vee$	disjunction	'... or ...'
$\rightarrow$	conditional	'If ... then ...'
$\leftrightarrow$	biconditional	'... if and only if ...'

Sentential Connectives

It is not the case that P	$\neg P$
P or Q	$(P \vee Q)$
P and Q	$(P \,\&\, Q)$
If P, then Q	$(P \rightarrow Q)$
P if and only if Q	$(P \leftrightarrow Q)$

Further symbolisation help:

Neither P nor Q	$\neg(P \vee Q)$ or $(\neg P \,\&\, \neg Q)$
P but Q	$(P \,\&\, Q)$
P unless Q	$(P \vee Q)$
P only if Q	$(P \rightarrow Q)$

Predicates

All Fs are Gs	$\forall x(Fx \rightarrow Gx)$
Some Fs are Gs	$\exists x(Fx \,\&\, Gx)$
Not all Fs are Gs	$\neg\forall x(Fx \rightarrow Gx)$ or $\exists x(Fx \,\&\, \neg Gx)$
No Fs are Gs	$\forall x(Fx \rightarrow \neg Gx)$ or $\neg\exists x(Fx \,\&\, Gx)$

Identity

Only c is G	$\forall x(Gx \rightarrow x{=}c)$ or perhaps $\leftrightarrow$.
Everything besides c is G	$\forall x(\neg\, x{=}c \rightarrow Gx)$
The F is G	$\exists x(Fx \,\&\, \forall y(Fy \rightarrow x{=}y) \,\&\, Gx)$
It is not the case that the F is G	$\neg\exists x(Fx \,\&\, \forall y(Fy \rightarrow x{=}y) \,\&\, Gx)$
The F is non-G	$\exists x(Fx \,\&\, \forall y(Fy \rightarrow x{=}y) \,\&\, \neg Gx)$

3.4 Using identity to symbolize quantities

There are at least _____ Fs.

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

There are at most _____ Fs.

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

one: $\forall x_1\forall x_2\big[(Fx_1 \,\&\, Fx_2) \rightarrow x_1{=}x_2\big]$
two: $\forall x_1\forall x_2\forall x_3\big[(Fx_1 \,\&\, Fx_2 \,\&\, Fx_3) \rightarrow (x_1{=}x_2 \vee x_1{=}x_3 \vee x_2{=}x_3)\big]$
three: $\forall x_1\forall x_2\forall x_3\forall x_4\big[(Fx_1 \,\&\, Fx_2 \,\&\, Fx_3 \,\&\, Fx_4) \rightarrow$
$(x_1{=}x_2 \vee x_1{=}x_3 \vee x_1{=}x_4 \vee x_2{=}x_3 \vee x_2{=}x_4 \vee x_3{=}x_4)\big]$
n: $\forall x_1 \ldots \forall x_{n+1}\big[(Fx_1 \,\&\, \ldots \,\&\, Fx_{n+1}) \rightarrow (x_1{=}x_2 \vee \ldots \vee x_n{=}x_{n+1})\big]$

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: $\exists x\big[Fx \,\&\, \forall y(Fy \rightarrow x{=}y)\big]$
two: $\exists x_1\exists x_2\big[Fx_1 \,\&\, Fx_2 \,\&\, \neg x_1{=}x_2 \,\&\, \forall y\big(Fy \rightarrow (y{=}x_1 \vee y{=}x_2)\big)\big]$
three: $\exists x_1\exists x_2\exists x_3\big[Fx_1 \,\&\, Fx_2 \,\&\, Fx_3 \,\&\, \neg x_1{=}x_2 \,\&\, \neg x_1{=}x_3 \,\&\, \neg x_2{=}x_3 \,\&$
$\forall y\big(Fy \rightarrow (y{=}x_1 \vee y{=}x_2 \vee y{=}x_3)\big)\big]$

$$n\text{:}\ \exists x_1 \ldots \exists x_n \big[F x_1 \ \& \ldots \& \ F x_n \ \& \ \neg x_1 = x_2 \ \& \ldots \& \ \neg x_{n-1} = x_n \ \& \\ \forall y \big(F y \rightarrow (y = x_1 \vee \ldots \vee y = x_n)\big)\big]$$

3.5 Basic deduction rules for TFL

Conjunction

$$
\begin{array}{r|ll}
m & X & \\
n & Y & \\
 & X \,\&\, Y & \&\text{I } m, n
\end{array}
$$

$$
\begin{array}{r|ll}
m & X \,\&\, Y & \\
 & X & \&\text{E } m
\end{array}
$$

$$
\begin{array}{r|ll}
m & X \,\&\, Y & \\
 & Y & \&\text{E } m
\end{array}
$$

Disjunction

$$
\begin{array}{r|ll}
m & X & \\
 & X \lor Y & \lor\text{I } m
\end{array}
$$

$$
\begin{array}{r|ll}
m & X & \\
 & Y \lor X & \lor\text{I } m
\end{array}
$$

$$
\begin{array}{r|ll}
m & X \lor Y & \\
i & \quad \vert\ \underline{X} & \\
 & \quad \vert\ \vdots & \\
j & \quad \vert\ Z & \\
k & \quad \vert\ \underline{Y} & \\
 & \quad \vert\ \vdots & \\
l & \quad \vert\ Z & \\
 & Z & \lor\text{E } m, i\text{–}j, k\text{–}l
\end{array}
$$

Conditional

$$
\begin{array}{r|ll}
m & \quad \vert\ \underline{X} & \\
 & \quad \vert\ \vdots & \\
n & \quad \vert\ Y & \\
 & X \to Y & {\to}\text{I } m\text{–}n
\end{array}
$$

$$
\begin{array}{r|ll}
m & X \to Y & \\
n & X & \\
 & Y & {\to}\text{E } m, n
\end{array}
$$

Contradiction

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

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

Negation

$$
\begin{array}{r|l l}
m & \quad\begin{array}{|l} X \\ \hline \vdots \\ \bot \end{array} & \\
n & & \\
 & \neg X & \neg\mathrm{I}\ m\text{–}n
\end{array}
$$

$$
\begin{array}{r|l l}
m & \quad\begin{array}{|l} \neg X \\ \hline \vdots \\ \bot \end{array} & \\
n & & \\
 & X & \mathrm{PbC}\ m\text{–}n
\end{array}
$$

Reiteration

$$
\begin{array}{r|l l}
m & X & \\
 & X & \mathrm{R}\ m
\end{array}
$$

Law of Excluded Middle

$$
\begin{array}{|l l}
X \vee \neg X & \mathrm{LEM}
\end{array}
$$

3.6 Derived rules for TFL

Disjunctive syllogism

$$\begin{array}{l|ll} m & X \vee Y & \\ n & \neg X & \\ & Y & \text{DS } m, n \end{array}$$

$$\begin{array}{l|ll} m & X \vee Y & \\ n & \neg Y & \\ & X & \text{DS } m, n \end{array}$$

Modus Tollens

$$\begin{array}{l|ll} m & X \to Y & \\ n & \neg Y & \\ & \neg X & \text{MT } m, n \end{array}$$

Double-negation elimination

$$\begin{array}{l|ll} m & \neg\neg X & \\ & X & \text{DNE } m \end{array}$$

De Morgan Rules

$$\begin{array}{l|ll} m & \neg(X \vee Y) & \\ & \neg X \,\&\, \neg Y & \text{DeM } m \end{array}$$

$$\begin{array}{l|ll} m & \neg X \,\&\, \neg Y & \\ & \neg(X \vee Y) & \text{DeM } m \end{array}$$

$$\begin{array}{l|ll} m & \neg(X \,\&\, Y) & \\ & \neg X \vee \neg Y & \text{DeM } m \end{array}$$

$$\begin{array}{l|ll} m & \neg X \vee \neg Y & \\ & \neg(X \,\&\, Y) & \text{DeM } m \end{array}$$

3.7 Basic deduction rules for FOL

Universal elimination

$$\begin{array}{r|ll} m & \forall x X(\ldots x \ldots x \ldots) & \\ & X(\ldots c \ldots c \ldots) & \forall\text{E } m \end{array}$$

Universal introduction

$$\begin{array}{r|ll} m & X(\ldots c \ldots c \ldots) & \\ & \forall x X(\ldots x \ldots x \ldots) & \forall\text{I } m \end{array}$$

c must not occur in any undischarged assumption
x must not occur in $X(\ldots c \ldots c \ldots)$

Existential introduction

$$\begin{array}{r|ll} m & X(\ldots c \ldots c \ldots) & \\ & \exists x X(\ldots x \ldots c \ldots) & \exists\text{I } m \end{array}$$

x must not occur in $X(\ldots c \ldots c \ldots)$

Existential elimination

$$\begin{array}{r|lll} m & \exists x X(\ldots x \ldots x \ldots) & & \\ i & & \begin{array}{|l} X(\ldots c \ldots c \ldots) \\ \hline \vdots \\ Y \end{array} & \\ j & & & \\ & Y & & \exists\text{E } m, i\text{–}j \end{array}$$

c must not occur in
any undischarged assumption,
in $\exists x X(\ldots x \ldots x \ldots)$,
or in Y

Identity introduction

$$\begin{array}{|ll} c = c & =\text{I} \end{array}$$

Identity elimination

$$\begin{array}{r|ll} m & a = b & \\ n & X(\ldots a \ldots a \ldots) & \\ & X(\ldots b \ldots a \ldots) & =\text{E } m, n \end{array}$$

$$\begin{array}{r|ll} m & a = b & \\ n & X(\ldots b \ldots b \ldots) & \\ & X(\ldots a \ldots b \ldots) & =\text{E } m, n \end{array}$$

3.8 Derived rules for FOL

$$
\begin{array}{l|ll}
m & \forall x \neg X & \\
 & \neg \exists x X & \text{CQ } m
\end{array}
\qquad\qquad
\begin{array}{l|ll}
m & \exists x \neg X & \\
 & \neg \forall x X & \text{CQ } m
\end{array}
$$

$$
\begin{array}{l|ll}
m & \neg \exists x X & \\
 & \forall x \neg X & \text{CQ } m
\end{array}
\qquad\qquad
\begin{array}{l|ll}
m & \neg \forall x X & \\
 & \exists x \neg X & \text{CQ } m
\end{array}
$$

In the Introduction to his volume *Symbolic Logic*, Charles Lutwidge Dodson advised: "When you come to any passage you don't understand, *read it again*: if you *still* don't understand it, *read it again*: if you fail, even after *three* readings, very likely your brain is getting a little tired. In that case, put the book away, and take to other occupations, and next day, when you come to it fresh, you will very likely find that it is *quite* easy."

The same might be said for this volume, although readers are forgiven if they take a break for snacks after *two* readings.