ELEMENTS OF DEDUCTIVE LOGIC

Elements of Deductive Logic

AN INTRODUCTION TO FORMAL LOGIC
AND ELEMENTARY METATHEORY

WITH EXERCISES

Antony Eagle

University of Adelaide

This book is open source. XeLaTeX source code for the book is available from the book's GitHub repository, github.com/antonyeagle/edl, or by contacting me: antony.eagle@adelaide.edu.au.

Last updated 2017-06-05

Preface

This is a textbook covering the basics of formal logic and elementary metatheory. One distinguishing feature is that it has more emphasis on metatheory (particularly for sentential logic) than comparable introductory textbooks. Another distinguishing feature is that, while it is not primarily a book in philosophy of logic, the book does pay close attention to some foundational issues about meaning, consequence, and the philosophical rationale behind various metalogical results.

This book was originally written as a set of notes to accompany lectures in the introductory-intermediate logic course at the University of Oxford, *Elements of Deductive Logic*, but has evolved significantly since then. It is now suitable for independent study. In the classroom, I have used it as the basis for a second course in logic, particularly for slightly more mathematically inclined students, and it could serve as the basis for a useful bridging course to a more advanced third logic course that covers the material in Boolos *et al.* (2007).

Acknowledgements Thanks especially to Ofra Magidor and Andrew Bacon for helpful feedback. Thanks also to James Studd, Brian King, Iain Atkinson, Katy Moe, Daniel Hoek, Anastasiya Kravchuk, Matt Nestor, Bernhard Salow, Petra Staynova, Laura Castelli, and Syed Qader for comments and corrections. Several chapters of this book also formed the basis for part of the course *Logic II* at the

University of Adelaide in 2014; thanks to the students in that course for their feedback. And the final version of the book was road-tested in an honours seminar on formal methods in philosophy at the University of Adelaide in 2017: thanks to students in that course for their help in refining the materials, especially the sample answers to exercises.

Contents

Chapter 0

Introduction

Introductory logic courses serve generally to tell you how to *use* a new logical language. An introductory course will give you an introduction to the grammar, or *syntax*, of the new formal language, tell you how to interpret the grammatical sentences by giving a *semantics*, and most importantly from the perspective of applications, tell you how to translate from natural languages into the formal language. With a semantics and an translation scheme, you can now use a formal language to do various things more easily than you could otherwise: you can express claims precisely and without risk of ambiguity (No problem with 'everybody loves somebody' in formal languages!), and check the validity or otherwise of arguments of certain forms, which – if you've done your translation well – reflects on the validity or otherwise of certain informal arguments you might be interested in as a philosopher or mathematician. If you have a formal proof system, giving rules for formal derivations in your languages, you can even construct arguments directly in the formal language in a more or less mechanical fashion, thus enabling you to prove various claims directly without having to detour through semantics.

Thinking about this for a second, though, we need some reassurance that the formal languages will do what we want them to. Is it true that if a formal proof is correct, then the argument it formalises is semantically valid? Is it the case that every valid argument has a corresponding formal derivation? Which things can and cannot be expressed in a given formal language? Is a formal language

adequate to each argumentative use we might have for it in mathematics, or in philosophy? Are formal derivations really purely mechanical, or must there be human judgment involved too? These are questions *about* formal logical languages, questions in what we might call the *metatheory* of logic. Such questions are the focus of this book. Such questions aren't about how to use formal logics to do things, but about whether they are fit for purpose, whether they can in fact do the things we want them to do.[1]

Answering such questions involves drawing a distinction between the *object language*, the one we are talking about, and the *metalanguage*, the language we are using to talk about the object language. For this book, the object languages are various more-or-less orthodox languages of classical logic: a sentential language $\mathscr{L}_1$,[2] a predicate language $\mathscr{L}_2$, and a predicate language with identity $\mathscr{L}_=$. And the metalanguage is the mathematically enriched version of English we use to talk about formal languages.

The Structure of this Book This book is divided into several parts.

I begin, in Part I (chapters 1 and 2), by outlining some mathematical tools – some elementary *set theory*, including the theory of relations and functions, as well as some remarks on the nature of mathematical proof – that will help us study and understand the formal languages which are our subject.

In Part II, I turn to sentential logic. In chapters 3 and 4, I briefly review the syntax and semantics of classical sentential logic. Then we'll start proving some results in metatheory. We'll prove some small but still interesting results about the truth-functions, and about the intimate connections between conjunction ('$\wedge$'), disjunction ('$\vee$'), and negation ('$\neg$'). In chapter 5, we'll prove our first 'big' result: the so-called *compactness* theorem for sentential logic. I also mention issues of computational implementation, in particular, the *decidability* of sentential logic.

We'll take a break from semantics-led approaches in Part III (chapters 6–9), to introduce and talk about *formal derivations* in sentential logic, and introduce

[1] Such questions could easily be asked about natural languages too, though since the syntax and semantics of natural language are orders of magnitude harder to specify precisely, it is correspondingly harder to establish anything conclusively about the features of such languages.

[2] This is sometimes called the language of *propositional logic*, but as I take propositions to be the meanings of sentences, and logic to be the study of consequence relations between meaningful sentences, I wish to avoid the terminology of 'propositional logic' as potentially misleading.

two kinds of derivation system: first tableaux, then natural deduction. We'll prove two central results connecting derivability with validity: the *soundness* and *completeness* theorems for sentential logic. We prove them for each of our derivation systems.

Part IV turns to a review of *predicate* logic and (in chapters 11 and 12) some of its derivation systems, and we'll prove soundness for standard predicate logic. I sketch proofs of completeness, and briefly consider issues of decidability. In chapter 13, issues about predicate logic with identity are discussed, including the soundness of the derivation systems we construct, and issues about numerical quantification and compactness are discussed.

In the final part (V), we turn to issues beyond classical logic, particularly picking up on some of the philosophical issues raised in earlier parts. The focus in chapters 15 and 16 is on the relation between logical languages and natural languages, particularly focussing on issues about conditionals, entailment, designation and the theory of relations.

Exercises and Further Reading This book also includes exercises, contained in a section entitled 'Exercises' at the end of each chapter. These exercises are mostly review and consolidation of the material in the chapter, but some are intended to extend and stretch a student's knowledge of the material. Solutions to selected exercises – some rather more detailed than others – are provided in Appendix A.

Many chapters also contain a section of 'Further Reading', which contains references to books and articles that may provide additional information on particular topics of interest. Details of the items there referred to can be found in the Bibliography at the end of the book.

Other Appendices The text makes free use of Greek letters as variables of various kinds; a quick refresher of the English names of these letters can be found in Appendix B. There are a number of defined expressions; a list of definitions can be found in Appendix B.

Some Other Useful Readings The languages and derivation systems used in this book are those developed in (Halbach, 2010) and . While the present book is self-contained, it may be helpful to review the material in Halbach's book as a useful

adjunct. Other useful sources of related material, that I drew on in a number of places in this book, are Beall and van Fraassen (2003); Boolos *et al.* (2007); Bostock (1997); Priest (2008) and Sider (2010).

Part I

Mathematical Preliminaries

Chapter 1

Mathematical Induction and The Notion of Proof

1.1 Inductive Proofs

Mathematical Proof What is a proof? A proof of a claim is just an argument, from assumptions that are known truths, that the claim is true. In a proof, the argument must be a *conclusive* one: if the assumptions are true, the claim must also be true. An argument involved in a proof is *necessarily truth-preserving*, so that in every possible scenario where the assumptions – also known as *premises* – are true, the claim being argued for – also known as the *conclusion* – is also true in that scenario.[1]

Such arguments are hard to come by. So we tend to find proofs in mathematics and the mathematicised areas of other disciplines such as physics, computer science, economics, and even philosophy. Other disciplines involve plenty of conclusive arguments, but the stumbling block tends to be finding starting assumptions which are known to be true. In the mathematical case, we begin our proofs ultimately with the axioms of a particular mathematical theory: necessary mathematical truths that are collectively sufficient, when supplemented by appropriate

[1] The conclusion need not be necessary itself: what is necessary is the conclusive relationship between the premises and conclusion.

definitions, to characterise a given body of mathematical truths. (Some might even wonder whether we know these axioms, but I'll set aside such scepticism in this book.) There are many issues surrounding the nature of proof that we will neglect; but basically, a proof is an absolutely *conclusive* argument for a given claim, based on assumptions which are themselves known.

We will offer a considerable number of proofs in this book. They are largely proofs about particular formal logical systems. (We will also look at the formal analogue of mathematical proofs carried out in such logical systems, which we call *deductions*, in chapter 6 and chapter 12.) In this chapter, we look informally at some of the techniques we will use in constructing these proofs.

Mathematical Induction We shall often use a powerful method of reasoning known as proof by *mathematical induction*.

The basic idea is simple. Assume we have a numbered sequence of things. Suppose (i) we can show that: the first member of the sequence has some property, and (ii) we can show that: if a member of the sequence has the property, then the immediately subsequent member also has the property. Then we can conclude that *every* member of the sequence has the property.

The easiest case is where the sequence *is* the natural (counting) numbers $\mathbb{N} = 0, 1, 2, 3\ldots$, but the principle also works for sequences of other things that are ordered like the natural numbers. So suppose we rank the children in a classroom from tallest to shortest. A ranking just is a way of associating each member of some collection with an initial fragment of the natural numbers. Suppose the tallest child is shorter than 2m. By the way we've constructed the ranking, if some randomly selected child is shorter than 2m, then the next child in the ranking (if there is one) is also shorter than 2m. So, we can conclude, every child in the class is shorter than 2m. Perhaps we do not always – or ever! – go explicitly through such reasoning in such mundane cases. But we can see in the example that the reasoning does seem to be correct, and to underlie our informal and immediate conclusion that if the tallest student is shorter than 2m, then so is every student.

The Weak Principle of Induction There are a number of formulations of induction. This one follows our intuitions closely. It is a principle about the legitimacy of a certain pattern of reasoning.

Definition 1 (Weak Principle of Induction). Assume there is a sequence of items S, ordered by the natural numbers. If both of the following conditions hold,

1. P is true of the first member of S; and
2. if P is true of the n-th member of S, then P is true of $n+1$-th member of S;

then, P is true of every member of S.

To apply this principle, we first establish the *base case* for condition (i); and then the *induction case*, where we assume that (for some *arbitrary* n) P holds of the n-th member, and show it holds of the $n+1$-th member, to show condition (ii). (Condition (ii) is a conditional claim: it can hold even if nothing is F, as long as, if any n is F, that suffices to establish that the subsequent thing is also F.) Having established those conditions, the principle of induction says, it is now legitimate to infer that every member of the sequence has the property in question. Here is a more mathematical example.

Theorem 1
For every natural number n,

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2}. \tag{1.1}$$

Proof. We first prove the *base case*, that the property holds of 0 (the first member of $\mathbb{N}$). Easy: $0 = \frac{0\cdot(0+1)}{2} = \frac{0}{2} = 0$. (Usually the base case is the easy part.)

Now we wish to prove the *induction step*. That is, we assume the property holds of n, and show it must hold of $n+1$. So we assume: $0 + 1 + \ldots + n = (n(n+1))/2$.

$$(0 + 1 + \ldots + n) = \frac{n(n+1)}{2} \tag{1.2}$$

$$(0 + 1 + \ldots + n) + n + 1 = \left(\frac{n(n+1)}{2}\right) + n + 1 \tag{1.3}$$

$$= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} \tag{1.4}$$

$$= \frac{(n+2)(n+1)}{2} \tag{1.5}$$

$$= \frac{(n+1)((n+1)+1)}{2} \tag{1.6}$$

Equation (1.6) is clearly just an instance of Equation (1.1) with '$n + 1$' in place of 'n', as required. This proves the induction step. □

The Strong Principle of Induction Here is another induction principle, which is sometimes easier to use than weak induction.

Definition 2 (Strong Principle of Induction). 'For all n, if for all $m < n$, $P(m)$ then $P(n)$' implies 'For all n, $P(n)$.'

To return to our schoolchildren example, we could reason informally using the strong principle like this. Pick any random child. We know that if each child earlier in the ranking than our chosen child is shorter than 2m, then our child too is shorter than 2m. Since our child was randomly chosen, we know that every child is such that, if all its precedessors in the ranking are shorter than 2m, they are too. We conclude that every child must be shorter than 2m.

The strong principle is just a conditional claim – there is no basis step. Why? Assume that for every n, if for each $m < n$ it is the case that $F(m)$, then $F(n)$. Consider, in particular, $n = 0$. Is it true that for all $m < 0$, $F(m)$? Well, if it were false, there would be a number less than 0 such that it isn't F. But there are no numbers less than zero, so in particular, there are none of them that are not F! So – in a degenerate or vacuous way – all numbers less than 0 are F. So the antecedent of the conditional is true, and since the conditional is true, the consequent must be true. The consequent is $F(0)$ – which is the basis step. Still, sometimes its useful to set out the basis step even when we're using strong induction.

The choice of whether to use the weak principle or the strong principle is stylistic, for they are equivalent in the sense that any proof using one could be in principled reformulated to use the other. I show one direction of the equivalence here.

Theorem 2 (Weak to Strong)
The weak principle of induction validates reasoning in accordance the strong principle of induction.

Proof. We assume the conditional premise of the strong principle: that for every n, if for all $m < n$ $F(m)$, then $F(n)$. We will also assume the weak principle of induction. And we'll prove from those two assumptions that every n is F. That

is, assuming the validity of reasoning in accordance with the weak principle of induction, we can demonstrate the validity of reasoning in accordance with the strong principle.

Here's how. We define a new condition, G, as follows:

$$G(n) =_{\text{df}} \text{for all } m < n,\ F(m).$$

(So a number has G if all its predecessors have F.) We can re-write our conditional assumption using G as follows:

(†) For every n, if $G(n)$ then $F(n)$.

Base case: we show that $G(0)$. Trivial, by the argument just given.

Induction step: we want to show that, on the assumption that $G(n)$, it follows that $G(n+1)$. Assume $G(n)$. By (†), $F(n)$. By the definition of G, for all $m < n$, $F(m)$. So for all $m \leqslant n$, $F(n)$. But it is obvious that the collection of numbers less than or equal to n, is the same as the collection of numbers less than $n+1$. So we have shown that for all $m < n+1$, $F(n)$. But by the definition of G again, we've just shown $G(n+1)$.

Applying the weak principle of induction to this base case and this induction step, we conclude that for all n, $G(n)$. By (†) again, for all n, $F(n)$, as desired. □

The Least Number Principle Here's another induction principle, of a different form to those we've just considered.

Definition 3 (Least Number Principle). If M is a subset of $\mathbb{N}$, and is non-empty, then M has a least member.

On the assumption that reasoning in accordance with the LNP is good, then reasoning in accordance with the weak principle is also good, as is shown by the following theorem.

Theorem 3 (LNP to Weak)
The weak principle of Induction is validated by the least number principle.

Proof. Assume that P is a property such that $P(0)$, and for every n, $P(n) \rightarrow P(n+1)$. We prove from the LNP and these assumptions that for every n, $P(n)$ (which is the weak principle).

Let M be the set of numbers which do *not* satisfy P. By the LNP, M has a least member if it is not empty – call this member m. $P(0)$ holds, by assumption, so $m > 0$; and since m is the least member of M, $P(m-1)$. But since if $P(n)$ then $P(n+1)$, $P(m)$ follows from $P(m-1)$. That contradicts our assumption that $m \in M$. So there can be no least member of M. A set only has no least member if it has no members at all; so there is no number n such that $\neg P(n)$, hence for every number n, $P(n)$.

To show that all three principles of induction we've considered are equivalent, we'd now need to show that the LNP follows from the strong or weak principles of induction; this is left for an exercise.

1.2 Proof by *Reductio*

Another important proof strategy that you will commonly use is proof by *reductio ad absurdum* (literally, 'reduction to the absurd'). In mathematics, this is also commonly known as *proof by contradiction*. The idea of this proof is to show that some claim is true by showing that its negation cannot be true. In fact, we just used this proof strategy in the proof of Theorem 3. Here is another example.

Theorem 4
There is no largest prime number.

Proof. Suppose – for the sake of argument – that k is the largest prime. Take all the prime numbers less than or equal to k. Multiply all these numbers together, and add 1. The resulting number cannot be divided by k, nor by any of the primes smaller than k – for if you were to divide by any of those numbers, the remainder would be one. Therefore it is either itself prime, or (by the fundamental theorem of arithmetic) it is divisible by a prime number greater than k. In either case, there is a prime greater than k, thus refuting our initial supposition that there is a greatest prime. So that initial supposition must be false, and theorem holds. □

I have made the role of the supposition explicit – we show that an absurdity, or a flat out contradiction, follows from our assumption by impeccable conclusive reasoning. But the hallmark of impeccable conclusive reasoning is that, if you start with a truth, you must also end up with a truth. Since we have ended up with something bound to be untrue – an impossibility – we must not have started with a truth.

1.3 Intuitionism

The foregoing line of reasoning is very persuasive, and its use is pervasive throughout mathematics and in this book. However, some mathematicians and philosophers, beginning with the influential topologist Brouwer, have wondered whether this proof strategy is legitimate. Everyone agrees that, in showing that our initial supposition of 'not S' for some English sentence S is untrue, we have shown that its negation, 'not not S' is true. But these *intuitionists* say that 'not not S' is not equivalent to S. Why do they say this? It is because proof by contradiction permits us to show that there is an entity with a certain property (because to assume there is no such entity would lead to a contradiction), without telling us what that entity is. Such proofs are called *non-constructive*, for they do not construct the entity which has the property in question.[2]

Intuitionism on Mathematical Truth What is the interpretation of mathematical truth that motivates intuitionists to reject the equivalence of 'not-not-P' and 'P' for at least some sentence P? It is this: *mathematical truth is mathematical proof* (Shapiro, 2000). Mathematical objects don't just exist, out there in some Platonic heaven, waiting for facts about them to be discovered. They need to be brought into existence by the activity of a mathematician. But bringing something into existence is a positive act – simply showing that it would be absurd if something didn't exist, is not yet to construct it. So intuitionists offer a reinterpretation

[2]Consider, for example, the classical proof that there are irrational numbers a and b such that a^b is rational. We can show this by considering $q = \sqrt{2}^{\sqrt{2}}$, and show that either $q^{\sqrt{2}}$ is rational if q is irrational, or that q itself is rational – either way, there is a pair of irrational numbers (either $a = b = \sqrt{2}$, or $a = q, b = \sqrt{2}$) such that a^b is rational. But we haven't shown *which* of these numbers is in fact rational – we haven't constructed an example, we've merely shown that there must be an example. (In fact, q is irrational, so that is an actual constructed example that verifies the theorem.)

of the language of proof – English, in our case – to reflect their philosophical commitments.

- The absurd sentence $\perp$ is not constructible.
- A construction of 'P and Q' consists of a construction of P and a construction of Q.
- A construction of 'P or Q' consists of a construction of P or a construction of Q.
- A construction of 'If P then Q' is a technique which transforms any construction of P into a construction of Q.

They define 'not-P' to mean: 'If P then $\perp$', i.e., there is a technique which turns a construction of P into a construction of the absurdity. 'not not P', accordingly, says that there is a technique which turns a technique turning a construction of P into the construction of the absurdity, itself into a construction of an absurdity. But that is not at all obviously the same as a construction of P.

The Law of Excluded Middle Characteristically, the rejection of the so-called *law of double negation elimination* leads to the rejection of the *law of excluded middle* (LEM): intuitionists reject the principle that for any P, either P or not P. For them, that is equivalent to: for any P, there is (already) either a construction of P, or a construction turning a construction of P into the construction of an absurdity. Why should we think, prior to any such construction being offered, that they antecedently exist?

We can however prove something closely related to LEM.

Theorem 5 (LEM–)
For any P, not-not-(P or not-P)

Proof. Assume (i) that for some P, it is the case neither that P, nor that not-P. Assume (ii) P: it follows that P or not-P, so our assumption (ii) must be false. If assumption (ii) were correct, in conjunction with assumption (i), we'd be able to construct a contradictory claim of the form Q and not-Q. That is absurd, and not constructible. So, under assumption (i) alone, we can turn any construction of P into a construction of the absurd, which is of course a construction of not-P

from assumption (i). But it follows from not-P, again trivially, that 'P or not-P'. So we can turn an assumption of (i) into a construction of the absurdity, which is a construction of not-not-(P or not-P). □

So we can show that it is absurd to deny LEM; but that is not yet tantamount to its being legitimate to assert it.

Resisting Intuitionism To insist on constructive proofs, and to eschew the use of proof by contradiction, is to cripple mathematics. This would be acceptable – perhaps – if there were a powerful philosophical rationale for thinking that, in general 'not not S' is not equivalent to S. But no persuasive examples have really been given, despite the best efforts of intuitionists to defend their view. So in this book we will adopt the standard highly plausible classical position, that proof by contradiction is perfectly acceptable.

Further Reading

On mathematical induction, see Machover (1996). Two philosophical attempts to motivate intuitionism are by Heyting (1956) and Dummett (1983). A useful discussion of the non-standard features of intuitionistic mathematics is by Iemhoff (2015).

Exercises

1. Prove that the strong principle of induction entails the LNP.
2. *Bivalence* is the principle that every meaningful declarative sentence is either true or false – that there are exactly two 'truth values'. What is the relationship between Bivalence and the Law of Excluded Middle?

Answers to selected exercises on page 237.

Chapter 2

Set Theory

2.1 Sets

A *set* is a collection of objects. Like a collection, it is a single thing, not a plurality. If X is a set, the objects in X are called its *elements* or *members*. We write $x \in X$ for 'x is an element of X' (and $y \notin X$ for 'y is not an element of X').

Sets are *individuated* by their members; or as we sometimes say, sets are individuated purely *extensionally*. There must be something to a set over and above its members – otherwise it would be simply a plurality identical to those members. But whatever that 'something' is, it is minimal enough that there cannot be two distinct sets which share the same members.Therefore, if X and Y name sets which have all and only the same members, $X = Y$. So we can specify a set by listing the members, conventionally in curly braces: $X = \{x_1, ..., x_n, \dots\}$.

In this, sets differ from *sequences*: two distinct sequences can have the same members as long as they occur in a different order in the two sequences. Some say that sets also differ from wholes in this respect: while you can make only one set from a given collection of members, you can make more than one object from a given collection of parts, if you arrange those parts in different ways. (This last example is however controversial: some think that appearances here are deceptive.)

A set is not *composed* of its members in any straightforward sense: it is not

a whole with its members as parts. We can see this most easily in the case of a *singleton* set, a set with just one member. Clearly $x \in \{x\}$; but since x itself need not be a set, $x \notin x$. Since sets are individuated by their members, and since $\{x\}$ has a member that x does not, $x \neq \{x\}$. So a set is not to be identified with the fusion of its members, even when that latter object exists.

Axioms The standard approach to set theory is to take this informal conception of a set as a collection, and set down *axioms* that characterise the behaviour of sets. We've already seen one of the standard axioms of so-called Zermelo-Fraenkel set theory: the axiom of *extensionality*:

Axiom 1 (Extensionality). For any sets X and Y, if for all objects x, $x \in X$ iff $x \in Y$, then $X = Y$.

I will below mention various axioms, the basic starting points of modern set theory, but we will largely proceed in an informal way. In particular, I will not be offering proofs from the axioms of all the set theoretic principles I state below.

Sets in Sets One thing is worth being explicit about at the outset. *Sets can have other sets as members.* We might be used to thinking of collections as collections of material objects (rocks, china plates, etc.) But since a set is not identical to its members, and we can talk about and quantify over sets, they are also objects available to be the members of further sets. We might *start* with the non-sets, or *ur-elements* as they are sometimes known; but once we make sets of them, we can then make further sets which include sets as members, and then further sets involving sets of sets, etc. This is the so-called *iterative conception of a set* (Boolos, 1971).

Defining Sets We've seen one way to define a set: list the members. But what if there are too many to list?

One thing we can do is define sets *inductively*. This means: given a base case, and a *generating relation*, we define a set that includes the base case, and anything related to a member of the set by the generating relation, and nothing else (the *closure condition*). Here's an example.

Definition 4 (Your Ancestors). *Base case:* Your mother and father are ancestors.
Generating Relation: Any parent of an ancestor is an ancestor.
Closure Condition: Nothing else is an ancestor of you.

Let A_{ae} be the set of my ancestors. Any member of A_{ae} is a parent of a parent of...a parent of me, with $n > 0$ occurences of 'a parent of'. The property 'is an ancestor of AE' is called the *ancestral* of the property 'is a parent of AE'. And more generally we can say that the relation R 'x is an ancestor of y' is the ancestral of the relation 'x is a parent of y'. Note that the set $A_{ae} = \{x : R(x, ae)\}$

Alternatively, we can specify a set by stating a condition F that the members all satisfy, typically in some formal mathematical language. Then we can specify the set of things x such that each x is F. We write this $\{x : F(x)\}$. If $Y = \{x : F(x)\}$, then $y \in Y$ iff $F(y)$. But in fact any collection of items forms a set: there needn't be a natural condition all the members satisfy.

Not just any condition will define a set. Consider the condition $F = x \notin x$. This defines a set $\mathbf{R} = \{x : x \notin x\}$, the set of all non-self-membered things.

Is $\mathbf{R}$ a member of $\mathbf{R}$? Assume $\mathbf{R} \in \mathbf{R}$. Then $\mathbf{R}$ must meet the defining condition F, so that $\mathbf{R} \notin \mathbf{R}$. But since $\mathbf{R}$ cannot both be and not be a member of itself, the assumption must be wrong. So $\mathbf{R} \notin \mathbf{R}$. But then $\mathbf{R}$ does meet F, and therefore $\mathbf{R} \in \mathbf{R}$. A contradiction follows. This is known as *Russell's paradox*.

The problem is assuming that *any* condition at all defines a set. The alternative is simple: to assume that, *if we are given a set X*, then any condition at all will *separate* the members of X into those which meet the condition and those which do not. That is: for any set X, and any condition F, there will exist a set Y which contains everything which is both a member of X and satisfies F. This is the axiom of *separation*, so called because it separates out those members of a set which have a certain specified feature.

Axiom 2 (Separation). If X is a set that is already given, and F is any condition, then there exists a set Y consisting of all the members of X which meet the condition F. (This may be trivial, in that all or no members of X meet the condition.) That is, for any X and F, this specification is well-defined and guaranteed to denote a set: $\{x \in X : F(x)\}$.

Relations Between Sets Some important relations of inclusion and exclusion between sets can be defined, once we are assured that some sets exist.

Definition 5 (Subset). A set X is a *subset* of Y, written $X \subseteq Y$ if every member of X is also a member of Y: for every x, if $x \in X$ then $x \in Y$. X is a *strict subset* (or *proper subset*) of Y if $X \subseteq Y$ and $Y \nsubseteq X$ (written '$X \subset Y$').

Trivially, for all sets X, $X \subseteq X$. It also follows that if $X \subseteq Y$ and $Y \subseteq X$ then $X = Y$ (easy). And also if $X \subseteq Y$ and $Y \subseteq Z$ then $X \subseteq Z$. Note that if F is some condition, the set $\{x : x \in X \text{ and } F(x)\}$, which is guaranteed to exist by the axiom of Separation, is always a subset of X.

Definition 6 (Superset). A set X is a *superset* of Y, written $X \supseteq Y$ if every member of Y is also a member of X: for every x, if $x \in Y$ then $x \in X$. X is a *strict superset* (or *proper superset*) of Y if $X \supseteq Y$ and $Y \not\supseteq X$ (written '$X \supset Y$').

Definition 7 (Disjoint). Sets X and Y are *disjoint* iff there is no x which is a member of both.

Operations on Sets The foregoing relations are defined between existing sets. But there are also generating operations, that given some existing sets yield some new sets. For each of these generating operations, we need to give a definition of the properties of the operation. But we also need an axiom to ensure that that operation succeeds, that is, that there is a set which satisfies the definition we have set down. The axiom of Separation supports our first two definitions:

Definition 8 (Intersection). If X and Y are sets, the set $X \cap Y$ which contains only members of both is called the *intersection.*

If $\mathbf{X} = X_1, \ldots, X_n, \ldots$ is a set of sets, $\bigcap \mathbf{X} = X_1 \cap \ldots \cap X_n \cap \ldots$.

Definition 9 (Relative Complement). If X and Y are sets, then $X \setminus Y$ is the set of all members of X which are *not* members of Y, called the *relative complement* of Y with respect to X.

Because both of these definitions involve taking a set that is already given and constructing a subset, the existence of the resulting sets is supported by the axiom of Separation. In the case of intersection, if X and Y exist, $X \cap Y$ exists: $X \cap Y =$

$\{x \in X : x \in Y\}$. Alternatively, we can denote this set as the set of any x that meets the conjunctive condition of being in both X and Y: $\{x : x \in X \text{ and } x \in Y\}$. In the case of relatively complement, if X and Y exist, $X \setminus Y$ exists: $X \setminus Y = \{x \in Y : x \notin Y\}$.

What if we want to make a set which isn't a subset of an already given set?

Definition 10 (Union). If X and Y are sets, then $X \cup Y$ (read 'the *union* of X and Y') is the set which contains all the members of X and all the members of Y.

If $\mathbf{X} = X_1, \ldots, X_n, \ldots$ is a set of sets, $\bigcup \mathbf{X} = X_1 \cup \ldots \cup X_n \cup \ldots$.

The existence of unions is supported by this axiom:

Axiom 3 (Unions). Suppose $\mathbf{X}$ is any set of sets (i.e., a set with only other sets as its members). For any such $\mathbf{X}$, there is a set Y such that, for any z, $z \in Y$ iff z is a member of one of the members of $\mathbf{X}$. In short: if X and Y exist, $X \cup Y$ exists: $X \cup Y = \{x : x \in X \text{ or } x \in Y\}$.

That is, Y is the set containing all the things which are in any of the sets in $\mathbf{X}$. If X and Y are sets, there is a set $\{A, B\}$.[1] It then follows from the axiom of Unions that there is a set which contains all the members of X and all the members of Y. By extensionality, there is just one such set; we call it $A \cup B = \bigcup\{A, B\}$.

Note the links between $\cap$ and 'and', and $\cup$ and 'or'.

The Empty Set If X is a set, then by the definition of relative complement, $X \setminus X$ is a set. Suppose $X \setminus X$ has a member, z. Then $z \in X \setminus X$ iff $z \notin X$ and $z \in X$. Contradiction; so $X \setminus X$ has no members. This set is the *empty set*, written $\varnothing$. $Y \setminus Y$ is also empty; by extensionality, $X \setminus X = \varnothing = Y \setminus Y$; there is only one empty set.

The empty set is the collection which has nothing in it. It corresponds to the empty list. While the empty collection has no members, that doesn't mean that it itself is nothing: it is the unique thing which is a set and also has no members.

If X is any set, then for any x, if $x \in \varnothing$ then $x \in X$ – for the empty set has no members at all. By the definition of subset, therefore, $\varnothing \subseteq X$, for any X.

It is easy to show that when X and Y are disjoint, $X \cap Y = \varnothing$.

[1] This is in fact another axiom, the axiom of Pairing.

Absolute Complement and the Universal Set The empty set is the smallest possible set in terms of its members. Is there a largest set? If there is a *universal set* Ω – a set which has everything (every individual *and every set*) as a member – we could define something a bit closer to negation than relative complement. We could define: $-X = \Omega \setminus X$; it would be the set of all things which are not in X.

This is *not* possible. Suppose there were a universal set. Then the axiom of separation, that if X is a set then there is a set Y which is a subset of X and such that for all $x \in X$ if $F(x)$ then $x \in Y$, will entail a contradiction.[2] So there is no universal set. (Philosophical puzzle: how do we quantify over – talk about – all sets, as we have been doing, if there is no collection big enough to contain them all? How indeed can we even talk about *everything* if there are too many things to be collected together into a set?)

Power Set So far you might think that sets are just collections of individuals (i.e., non-sets). (Well, we've already mentioned the axiom of Pairing, so you probably don't think that.) But set theory is much more general than that: anything at all, including other sets, can be gathered into a set (as long as we don't try to gather too many things).

One useful set of sets is the *power set* of a set X. This is, intuitively, the set of all ways to choose some members from X. We can choose no members; we can choose all the members; we can choose some but not all. Each such way of choosing determines a subset of X, so the set of all ways to choose members from X is exactly the set of all subsets of X:

Definition 11 (Power Set). Given a set X, the *power set* of X, denoted $\wp(X)$, is defined: $\wp(X) = \{Y : Y \subseteq X\}$. (It is another axiom of set theory that for any set, its power set exists – the aptly named Power Set axiom.)

Notice that, for any X, $\varnothing \in \wp(X)$; and $X \in \wp(X)$. The name 'power set' comes from this fact:

Theorem 6

If X has n members, $\wp(X)$ has 2^n members.

[2] Hint for exercises: Apply the condition $x \notin x$ to the universal set Ω in the axiom of separation, and we will construct the Russell set $\mathbf{R}$ perfectly legitimately. Since the Russell set does not exist, something's gone wrong: the only candidate is the assumption that there is a set of everything.

Proof. A subset of X is a set containing perhaps some but not necessarily all the members of X. We can specify a subset by saying, of each member of X, whether it is in the subset. Suppose the members of X are $x_1, \ldots, x_n$. Then a finite binary sequence, consisting of 1s and 0s, of length n suffices to determine a subset of X, by the following principle: let x_i be a member of the subset associated with s iff the ith place in the sequence s is a 1. By extensionality, each subset of X corresponds to exactly one such sequence. So how many finite binary sequences of length n are there? Each place i in the sequence can be either 1 or 0, so there are 2^n such sequences. So there are 2^n distinct subsets of X, and hence 2^n members of $\wp(X)$.□

Ordered Pairs and Sequences As mentioned above, sets are identical iff they have the same members. How those members happen to be listed is irrelevant: $\{x, y\} = \{y, x\} = \{x, x, y\}$. So if we want to capture the idea of an *ordered sequence*, it cannot be a set – order doesn't matter for sets, and one can't have repeated elements in a set. Yet we can model, or represent, sequences just using sets.

Definition 12 (Ordered Pair). An *ordered pair* $\langle x, y\rangle$ is defined as the unordered set $\big\{x, \{x, y\}\big\}$.

This is not to say that ordered pairs are and always have been special kinds of sets. The existence of multiple equally good models for how to represent ordered pairs set-theoretically undermines any claim that ordered pairs are identical to sets. Kuratowski's original definition of an ordered pair was this different proposal:

Definition 13 (Kuratowski Ordered Pair). A *Kuratowski ordered pair* $«x, y» =_{\text{df}} \{\{x\}, \{x, y\}\}$.

If we take seriously the identification of ordered pairs with sets, we need to answer which of these definitions identifies the 'right' sets. But any such answer would be arbitrary, because it would involve picking one equally good representation of ordered pairs over another. The choice to model ordered pairs as in the text, or by Kuratowski's alternative, is a choice of model – it has no significance for what is being modelled, namely, sequences of length 2.

Why do I say this? Because we can show that both our definition and Kuratowski's manage to capture the behaviour of ordered pairs. We do so by establishing the fundamental principle of the identity of ordered pairs follows from

the definitions, namely, that ordered pairs are identical iff they have the same first member, and the same second member. What we show is that, if we define ordered pairs as above, they have all the characteristic features of ordered pairs.

Theorem 7 (Criteria of Identity for Ordered pairs)
$\langle x, y\rangle = \langle u, v\rangle$ *iff* $x = u$ *and* $y = v$.

Proof. $\langle x, y\rangle = \langle u, v\rangle$ iff $\{x, \{x, y\}\} = \{u, \{u, v\}\}$ iff (i) $x = u$ and $\{x, y\} = \{u, v\}$ or (ii) $x = \{u, v\}$ and $\{x, y\} = u$. Since (ii) is not possible – it would entail that x is a member of one of its own members – it must be that (i), which in turn holds iff $x = u$ and $y = v$ (we need extensionality again to show that if $\{x, y\} = \{x, v\}$ then $y = v$). □

In the proof of this theorem, we had to appeal to the fact that no set is a member of a member of itself. (In proving the parallel criteria of identity theorem for Kuratowski ordered pairs, we need not appeal to this fact, showing which I leave as an exercise.) This fact obviously holds on the iterative conception of a set. It tacitly relies on this characteristic axiom of Zermelo-Fraenkel set theory:

Axiom 4 (Foundation). Every non-empty set X contains a member y such that X is disjoint from y: $X \cap y = \varnothing$.

(This axiom is also known as *regularity*.)

Theorem 8
No set is a member of itself.

Proof. Let X be an arbitrary set. Clearly $X \in \{X\}$, which latter set exists by Pairing. By Foundation, since $\{X\}$ is non-empty, it has a member y such that $\{X\} \cap y = \varnothing$. Since the only member of $\{X\}$ is X, it must be that $\{X\} \cap X = \varnothing$. But if $X \in X$, then X and $\{X\}$ do have a member in common – namely, X – and their intersection isn't empty. So $X \notin X$. □

This theorem gives another route to avoiding Russell's paradox from earlier (page 17). But Foundation also enables us to rule out longer 'cycles' of inclusion, since it entails that there are no infinite descending chains of set membership: every set bottoms out eventually in some members of its members of its members… that have no members themselves.

We can extend the idea of an ordered pair to an ordered sequence of any number of elements, as follows:

Definition 14 (Ordered n-tuple). Let the notation $\langle\!\langle x_1, \ldots, x_n \rangle\!\rangle$ denote an ordered n-tuple, for $n \geqslant 2$. We offer an inductive definition.

- *Base case.* $\langle\!\langle x_1, x_2 \rangle\!\rangle =_{\text{df}} \langle x_1, x_2 \rangle$.
- *Inductive step.* $\langle\!\langle x_1, \ldots, x_{n+1} \rangle\!\rangle =_{\text{df}} \langle \langle\!\langle x_1, \ldots, x_n \rangle\!\rangle, x_{n+1} \rangle$.

From now on, I will abuse notation, and use '$\langle$' and '$\rangle$' to delimit all ordered sequences, not only pairs. But, speaking strictly, when in the following I denote an ordered triple by $\langle x, y, z \rangle$, that expression should be interpreted as $\langle\!\langle x, y, z \rangle\!\rangle = \langle \langle x, y \rangle, z \rangle$. It doesn't really matter *how* we define ordered tuples; all that matters is that they are well-defined set-theoretic entities which behave as expected. (So, strictly speaking, we'd need to prove a theorem analogous to Theorem 7 showing that ordered n-tuples as just defined behave in the right way. We are not going to do that.)

One remaining case deserves attention. Our notation covers all ordered sequences of two or more members. What about the degenerate case of an ordered sequence with just one member? We shall adopt the following harmless convention (which will however be of use in §10.2):

Definition 15 (Ordered 1-tuple). An ordered 1-tuple $\langle x \rangle$ is simply x itself.

This choice has the odd but harmless consequence that everything is a sequence.

2.2 Relations

Intuitively, a relation R characterises the nature of the connections between some things (its *relata*), or an aspect of the structure of some collection of things. We will model a relation set theoretically as a set of *sequences* of entities; we care not just about which things are involved in the relation, but which way they are related. (We need to be able to distinguish when Alice loves Bob from the case where Bob loves Alice, and simply saying that Alice and Bob are related by the '– loves –' relation doesn't allow us to do that.) So the set

$$\{\langle \text{Melbourne}, \text{Victoria} \rangle, \langle \text{Adelaide}, \text{South Australia} \rangle, \ldots\}$$

is (part of) the relation of ' – is the capital city of – '.

Definition 16 (Binary Relation). X is a binary relation on a *domain* D iff it is any set that contains nothing except ordered pairs $\langle x, y\rangle$ where $x \in D$ and $y \in D$. ($\varnothing$ contains nothing at all, and so nothing other than ordered pairs, so is a binary relation).

Definition 17 (Cartesian Product). The *Cartesian product* of sets X and Y, written '$X \times Y$', is the set of all pairs $\langle x, y\rangle$ such that $x \in X$ and $y \in Y$.

A binary relation on D is a subset of $D \times D$. A relation can be generalised to arbitrary numbers of relata:

Definition 18 (n-place Relation). X is an n-place relation on D iff it is any set of ordered n-tuples of members of D. An n-place relation is a subset of $\underbrace{D \times \ldots \times D}_{n}$.

This identification of relations with sets is not necessarily a 'metaphysical' hypothesis. Just as in the case of ordered pairs, we need not literally assert that relations are and always were sets. Rather, this is a model that can be used to represent some interesting features of relations. I say some more about the metaphysics of relations in chapter 13.

Properties of Binary Relations Binary relations have many interesting properties. We can often represent these properties *graphically*, if the domain on which the relation is defined is finite.

Definition 19 (Finite Directed Graph). A *finite directed graph* on a domain D is a pair $\langle V, A\rangle$ where the set of *vertices* V is D, D is finite, and A is a set of *arrows*, directed lines from members of V to members of V. (Not every member of V needs to have an arrow attached to it.)

A graph on D represents a binary relation R on D just in case

1. $\langle x, y\rangle \in R$ iff $x \in V$ and $y \in V$.
2. $\langle x, y\rangle \in R$ iff there is an arrow from x to y in A.

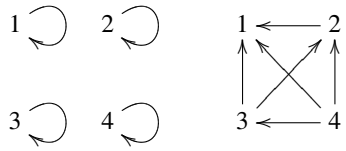


Figure 2.1: Reflexivity and Transitivity

Reflexivity, Transitivity, and Symmetry Consider the domain $D = \{1, 2, 3, 4\}$.

Definition 20 (Reflexive). A relation on D is *reflexive* iff for all $x \in D$, $\langle x, x \rangle \in R$.

Graphically, that means, as in the figure on the left in Figure 2.1, that each vertex has an arrow pointing to itself. A relation is *irreflexive* iff no vertex has an arrow pointing to itself.

Definition 21 (Transitive). A relation on D is *transitive* iff for all $x, y, z \in D$, if both $\langle x, y \rangle \in R$ and $\langle y, z \rangle \in R$, then also $\langle x, z \rangle \in R$.

Graphically, that means, as in the figure on the right in Figure 2.1, that whenever there is an indirect sequence of arrows between two vertices, there is also a 'shortcut'. A relation is *intransitive* iff there is never a shortcut.

Definition 22 (Symmetric). A relation R on D is *symmetric* iff whenever $\langle x, y \rangle \in R$, then $\langle y, x \rangle \in R$.

Graphically, this means (as on the left in Figure 2.2) that whenever there is an arrow from one vertex to another, there will be a return arrow. A relation is *asymmetric* iff there are no returning arrows at all.

Definition 23 (Anti-symmetric). A relation R on D is *anti-symmetric* iff whenever $\langle x, y \rangle \in R$ and $\langle y, x \rangle \in R$, then $x = y$ (sometimes this is known as *weak asymmetry*).

Graphically, this means (as on the right in Figure 2.2) that the only 'returning' arrows are loops. We need to clearly distinguish non-reflexivity from irreflexivity: the former is just the failure of reflexivity – i.e., at least one object in the domain doesn't bear R to itself – while the latter involves every object in the domain

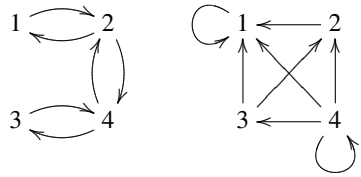


Figure 2.2: Symmetry

not bearing R to itself. Likewise, we need to distinguish non-transitivity from intransitivity, and non-symmetry from asymmetry *and* antisymmetry.

The observant reader will have noticed that, while we defined *reflexive on D*, *symmetric on D*, etc., the only definition which actually mentions D is the definition of reflexivity. All the other definitions are *conditional* in form: they say, if certain pairs are in the relation, then certain other pairs will be too. We needn't specify a domain to check whether these conditionals hold of any relation. But to check whether a relation is reflexive, we need not only the ordered pairs of the relation, but also what the domain is, so we can see if any members of the domain are missing from the relation. Reflexivity is an extrinsic property of a relation, because it is relative to the domain from which the relata of the relation may be drawn. The others are intrinsic properties of a relation; whether the relation has them is fixed by which pairs are in the relation. The very same set of pairs might be reflexive on one domain and non-reflexive on another, but it will be symmetric on every domain if it is symmetric on any.

Equivalence Relations

Definition 24 (Equivalence Relation). If R on D is reflexive, transitive, and symmetric, then it is an *equivalence relation.*

An equivalence relation divides up the domain into *equivalence classes*, every member of which bears the relation to every other member of it. The relation 'is the same height as' is an equivalence relation on the domain of all people; we say that it *induces a partition* on the domain of people, sorting them into groups which are uniform with respect to height.

The most obvious equivalence relation on a domain D is the identity relation, the set of all pairs $\langle x, x\rangle$ such that x *in* D. It is reflexive by definition, and

moreover symmetric and transitive; since there are *never* arrows from any x to any distinct y, it follows that every time there are such arrows, there are return arrows and shortcuts. (Notice that the identity relation is both symmetric and anti-symmetric.)

Any relation on the empty set is also trivially an equivalence relation. While every empty relation is trivially symmetric and transitive, it will generally fail to be reflexive unless the domain is empty too.

Seriality Reflexivity guarantees that every thing in the domain bears the relation to at least one thing, and moreover that among those things is the object itself. If a relation satisfies the first condition, though perhaps without meeting the second, it is *serial*:

Definition 25 (Serial). A relation on D is *serial* iff for all $x \in D$, there is some $y \in D$ such that $\langle x, y\rangle \in R$.

Graphically, that means, that each vertex has an arrow emerging from it. A relation is *non-serial* if at least one vertex has no arrow emerging from it. (The notion of an '*aserial*' relation – one where no vertex bears the relation to anything – is trivial, since only the empty relation could be aserial.) It is obvious that a reflexive relation is serial.

Theorem 9

A transitive, symmetric, and serial relation is reflexive (and hence is an equivalence relation).

Proof. Suppose R is transitive, symmetric, and serial on D. Pick an arbitrary member of D, x. By seriality, there is a $y \in D$ such that $\langle x, y\rangle \in R$. By symmetry, $\langle y, x\rangle \in R$. By transitivity, $\langle x, x\rangle \in R$. Since x was arbitrary, R is reflexive. □

Connectedness An equivalence relation divides up the domain into isolated groups of mutually interrelated elements. A connected relation, intuitively, is one where no item is isolated. But there are a number of things we could mean by this. First, we define a (directed) path:

Definition 26 (Path). There is a *path* from x to y in R iff there exists a sequence $z_1, \ldots, z_n$ such that $\langle x, z_1\rangle \in R$ and $\langle z_1, z_2\rangle \in R$ and ...and $\langle z_n, y\rangle \in R$.

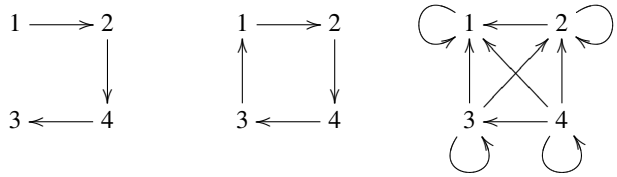


Figure 2.3: Connectedness and Totality

Now we may define:

Definition 27 (Strong Connectedness). R on D is *connected* iff for any $x \in D$ and $y \in D$, such that x and y are distinct, then there is a path from x to y in R.

This is illustrated in the middle graph in Figure 2.3, where by following the arrows one can get from any vertex to any other, perhaps by a number of intermediate vertices.

There is another property of binary relations, sometimes called *weak connectedness*, which holds iff for any nodes on the graph of R, x and y, there is some path that joins x to y moving only along arrows (perhaps in the 'backwards' direction). This is illustrated in the left hand graph in Figure 2.3. For any binary relation S, let S^* be the relation defined by $\langle x, y\rangle \in S^*$ iff $\langle x, y\rangle \in S$ or $\langle y, x\rangle \in S$.

Definition 28 (Weak Connectedness). A relation R is weakly connected on D iff R^* is connected on D.

There is an even stronger condition than connectedness, illustrated in the right hand graph in Figure 2.3:

Definition 29 (Totality). A relation R on D is *total* iff for any x and y in D, either $\langle x, y\rangle \in R$ or $\langle y, x\rangle \in R$.

Theorem 10
If R is total on D, then R is reflexive on D.

Proof. If R is total, then for any x and y in D, either $\langle x, y\rangle \in R$ or $\langle y, x\rangle \in R$. Suppose $x = y$; then either either $\langle x, x\rangle \in R$ or $\langle x, x\rangle \in R$. Therefore, for any $x \in D$, $\langle x, x\rangle \in R$, i.e., R is reflexive on D. □

Inverse and Complement Relations Given a relation R on D, we can define two interesting further relations in terms of it:

Definition 30 (Inverse). Given a relation R, its *inverse* R^{-1} is defined to be this set: $\{\langle y, x\rangle : \langle x, y\rangle \in R\}$.

Definition 31 (Complement). Given a relation R on D, its *complement* R' on D is defined to be this set: $(D \times D) \setminus R$ – i.e., the set of all ordered pairs of elements of the domain *not* in R.

Note that a complement relation is always with respect to the underlying domain – there is no such thing as *the* complement of a relation – whereas we can define an inverse relation just with reference to the original relation.

Theorem 11 (Property Preservation)

- *If R is reflexive on D then R^{-1} is reflexive on D and R' is irreflexive on D;*
- *if R is symmetric then R^{-1} and R' are symmetric;*
- *if R is transitive then R^{-1} is transitive;*
- *if R is connected on D then R^{-1} is connected on D.*

Proof. I prove one case: reflexivity. If R is reflexive on D, then for all $x \in D$, $\langle x, x\rangle \in R$. By definition of inverse, then, each $\langle x, x\rangle \in R^{-1}$; i.e., R^{-1} is also reflexive on D. By definition of complement, *no* $\langle x, x\rangle \in R'$; so R' is irreflexive on D. □

Orderings

Definition 32 (Ordering). A relation R on D is a

Partial order iff it is reflexive, transitive, and anti-symmetric (e.g., $\subseteq$ on a set of sets; see Figure 2.4);

Strict Partial Order iff it is transitive and asymmetric (i.e., $\subset$);

Total Order iff it is a total partial ordering (i.e., $\leqslant$ on $\mathbb{N}$). Also sometimes known as a *linear order*;

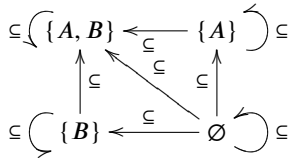


Figure 2.4: Partial ordering of a set of sets by $\subseteq$

Strict Total Order iff it is a total strict partial order (i.e., $<$).

As defined above, a total order is reflexive, transitive, antisymmetric, and total. By Theorem 10, a total order is reflexive. So we could have defined a total order as one that is total, transitive, and antisymmetric.

Well-Orderings Suppose R is a total order on D. If $\langle x, y\rangle \in R$, then we say x *precedes* y. If $\langle x, y\rangle \in R$ and $x \neq y$ and there does not exist a z such that $\langle x, z\rangle \in R$ and $\langle z, y\rangle \in R$, then x *immediately precedes* y. (We can give exactly similar definitions for *succeeds*.)

An element x of D is *minimal* iff there is no distinct y that precedes x. An element x is *least* if it precedes *every distinct* $y \in D$. (Corresponding definitions can be given for maximality and greatest.)

Definition 33 (Well-ordering). R is a well-ordering of D iff it is a total order and every non-empty subset d of D has a least member under R.

Example: $\leqslant$ is a well-ordering of the natural numbers $\mathbb{N}$. Here is another well-ordering of $\mathbb{N}$:

$$x \preccurlyeq y \quad \text{iff} \quad \begin{cases} x \text{ is even and } y \text{ is odd; or} \\ x \text{ and } y \text{ are even and } x \leqslant y\text{; or} \\ x \text{ and } y \text{ are odd and } x \leqslant y. \end{cases} \tag{2.1}$$

The order induced by $\preccurlyeq$ yields $0, 2, 4, \ldots, 1, 3, 5, \ldots$. In this ordering, every subset of $\mathbb{N}$ has a $\preccurlyeq$-least element. Notice, however, that 1 is not least, but it has no immediate predecessor.

The less-than-or-equal-to relation on natural numbers $\leqslant$ is a well-ordering of $\mathbb{N}$. It can be extended in an obvious way to the less-than-or-equal-to relation

on the positive and negative integers $\mathbb{Z}$, which we denote $\leqslant_{\mathbb{Z}}$. (This is strictly speaking a different relation, since $\langle -1, 0\rangle \in \leqslant_{\mathbb{Z}}$ but $\langle -1, 0\rangle \notin \leqslant$.) $\leqslant_{\mathbb{Z}}$ is a total or linear order on $\mathbb{Z}$, but is not a well-ordering, because the set of all negative integers is a non-empty subset of $\mathbb{Z}$ but has no least member under $\leqslant_{\mathbb{Z}}$ (it does have a greatest member though, unlike $\mathbb{N}$). But $\mathbb{Z}$ can be well-ordered: $x \leqslant_{\text{abs}} y$ iff either (i) $|x| \leqslant |y|$ or (ii) $|x| = |y|$ and $x \leqslant y$, where '$|x|$' here denotes the absolute value of x. This induces the order $0, -1, 1, -2, 2, \ldots$.

2.3 Functions

Consider the relation '– is the capital city of –', conceived of as relating Australian states to Australian cities. This relation has an interesting feature: if a state stands in the relation to a city, then it stands in the relation to exactly one city. This is in fact made explicit by the way we have specified the relation: a city C stands in the relation to a state S iff C is (is identical to) the capital city of S, where the descriptive noun phrase 'the capital city of S' manages to successfully denote something just in case there is only one capital city. Not all relations have this feature: 'x is taller than y', for example, cannot be formulated as an identity between x and something denoted by a descriptive phrase involving y.

Relations which have this special feature, where whenever xRy, then there is no distinct z such that xRz, are known as *functions*.

Definition 34 (Unary Function). If R is a binary relation such that for any x, y, z, if $\langle x, y\rangle \in R$ then $\langle x, z\rangle \in R$ iff y is identical to z, then R is a unary (one-place) function.

We write $f_R(x) = y$ iff $\langle x, y\rangle \in R$. The notation '$f_R(x)$' is a noun phrase, typically corresponding to a definite description like 'the capital city of x'. We say x is the *argument* given to f, and y the *value* of f given that argument. The fact that, for a given argument, a function has a unique value, justifies our talking of '*the* value of the function for a given argument', and justifies the link to natural language definite descriptions as corresponding to a function with an argument – see the discussion of definite descriptions in chapter 16.

The notion of a function can be generalised to an arbitrary n-place relation, which will give rise potentially to an $n-1$-place function.

Definition 35 (Function). f is an n-place function iff (i) it is a relation (ii) and for any sequence $x_1, \ldots, x_n$ of objects, and any y, z, if $\langle x_1, \ldots, x_n, y\rangle \in f$ and $\langle x_1, \ldots, x_n, z\rangle \in f$, then $y = z$ (i.e., each sequence of things stands in the relation to at most one thing).

Since a function is a relation, some of its properties are defined only relative to a domain. There is another notion, also confusingly called the domain of a function, which can be thought of as the things the function accepts as arguments.

Definition 36 (Domain). The *domain* (note: different to the domain of a relation) of a function f is the set

$$\mathscr{D}(f) = \{x : \text{there is a } y \text{ such that } \langle x, y\rangle \in f\}.$$

There is also the range of the function, which are the things the function can yield as values.

Definition 37 (Range). The *range* of f is the set

$$\mathscr{R}(f) = \{y : \text{there is a } x \text{ such that } \langle x, y\rangle \in f\}.$$

Definition 38 (Partial and Total). A function f is *partial* on domain D if $\mathscr{D}(f) \subset D$; it is *total* if $\mathscr{D}(f) = D$.

Obviously any function f is total on $\mathscr{D}(f)$, but it may be total on some domains and partial on others. In this sense, totality and partiality of a function is another extrinsic property of the underlying relation.

Operators

Definition 39 (Operator). If f is a function, and $\mathscr{R}(f) \subseteq \mathscr{D}(f)$, then f is an *operator*.

Consider the numerical function $\mathsf{sq} = \left\{\langle x, y\rangle : x^2 = y\right\}$, otherwise written $\mathsf{sq}(x) = x^2$. This generalises in the obvious way to binary functions: if the values of the function are appropriate arguments, it is an operator. An example is the addition function on the domain of integers, $\mathsf{plus} = \{\langle x, y, z\rangle : z = x + y\}$. This is an arithmetical operator.

Some properties of functions are conveniently illustrated by plus.

Definition 40 (Commutative). A binary function f is commutative iff

$$f(x, y) = f(y, x).$$

Definition 41 (Associative). A binary operator f is associative iff

$$f(x, f(y, z)) = f(f(x, y), z).$$

(We need f to be an operator to ensure that it is defined when its values are treated as arguments in this way.)

It is obvious that plus is both commutative and associative: $x+y = y+x$ and $x+(y+z) = (x + y) + z$. A commutative but not associative operator is $\mathsf{mean}(x, y) = \frac{x+y}{2}$. Clearly $\mathsf{mean}(x, y) = \mathsf{mean}(y, x)$. (This follows from the commutativity of plus.) But $\mathsf{mean}(x, \mathsf{mean}(y, z))$ is not in general equal to $\mathsf{mean}(\mathsf{mean}(x, y), z)$. An associative but non-commutative operator is the concatenation operator on strings $\frown$, which takes two strings and joins them into a longer string by appending the second to the first. Clearly $(x \frown y) \frown z = x \frown (y \frown z)$: 'te'$\frown$'a' = 'tea' = 't'$\frown$'ea'. But $x \frown y$ isn't necessarily $y \frown x$: 'tea' isn't 'ate'.

Properties of functions Suppose f is a function, and X and Y are some sets. There are some relational properties that f might have relative to X and Y.

Definition 42 (Into, Onto, etc.). **Into** f is a function from X *into* Y iff $\mathscr{D}(f) = X$ and $\mathscr{R}(f) \subseteq Y$.

Onto f is a function from X *onto* Y iff $\mathscr{D}(f) = X$ and $\mathscr{R}(f) = Y$. An *onto* function is also known as a *surjection.*

One-One f is a *one-one* function from X to Y iff f is into and whenever $x \neq y$, $f(x) \neq f(y)$. A one-one function is also known as an *injection.*

Bijection f is a *bijection* from X to Y (or, is a *one-one correspondence*) iff f is one-one and onto.

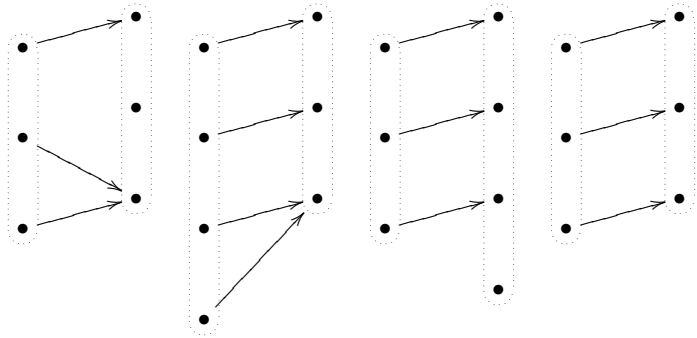


Figure 2.5: Properties of functions

It is important to remember as always that these features are always relative to the sets X and Y. A function that is a bijection between X and Y needn't be a bijection between X' and Y'. These properties are illustrated in Figure 2.5.

Some examples; these are all total functions from $\mathbb{N}$:

- The constant function $f(x) = 1$ is into $\mathbb{N}$.
- The function $g(x) = \begin{cases} x/2 & \text{if } x \text{ is even} \\ (x+1)/2 & \text{if } x \text{ is odd} \end{cases}$ is onto $\mathbb{N}$. (Every natural number is the value of this function for two arguments.)
- The function $h(x) = x^3$ is one-one from $\mathbb{N}$ to $\mathbb{N}$. (Every natural number has a unique cube, every cube has a unique cube root – but not every natural number is a cube.)
- The identity function $i(x) = x$ is a bijection from $\mathbb{N}$ to $\mathbb{N}$.

Because a function is a relation, it has an inverse and complemenent. The complement is not very interesting from the perspective of functions, since only in trivial cases will the complement of a function be a function. But the inverse of a function may be of more interest. The definition of an inverse relation in Into, Onto, etc. 30 suffices to define an inverse function, but here is an equivalent definition directly in terms of functions:

Definition 43 (Inverse). If g is a function, it is the *inverse* of f iff $\mathscr{D}(f) = \mathscr{R}(g)$, $\mathscr{R}(f) = \mathscr{D}(g)$, and for all x, $g(f(x)) = x$. In that case, we often write f^{-1} for g.

Obviously, f is also the inverse of f^{-1}.

Note that not every function has an inverse function (it always has an inverse relation, which may or may not be a function):

Theorem 12
A function has an inverse iff it is one-one.

Proof. Left for exercises. □

Theorem 13
If f is one-one but not a bijection, f^{-1} is a partial function.

Proof. If f is one-one from X to Y but not a bijection, then there exists a $y \in Y$ which is not the value of f for any member of X. Assume for *reductio* that f^{-1} is total on Y. Then there must be some $x \in X$ such that $f^{-1}(y) = x$. But then by the definition of inverse, $f(x) = f(f^{-1}(y)) = y$. But then x is a member of X which yields y as value when given to f as argument; contradiction. So f^{-1} is not total. □

2.4 Size

Definition 44 (Finite). A set X is finite iff there exists a set of natural numbers $N = \{1, \ldots, n\}$ and a bijection f from X to N.

A finite set can be put into one-one correspondence with some initial fragment of the sequence of natural numbers. (This corresponds to the notion, familiar from childhood, of counting some things by pointing to each of them while reciting some initial fragment of the natural numbers.) A set is infinite iff it is not finite. Where there is a bijection between X and $\{1, \ldots, n\}$, X has n members – we say that X has cardinality n, $|X| = n$.

Now we will define some further notions useful for talking about *size*.

Definition 45 (Enumeration). f is an *enumeration* of X iff $\mathscr{D}(f) \subseteq \mathbb{N}$ (the natural numbers), $\mathscr{R}(f) = X$, and f is onto.

An enumeration associates each member of X with some number. This is a generalisation of the ordinary notion of counting, however: since we only require that an enumeration be onto, all sorts of weird 'counting' procedures are included: e.g, the constant function from $\mathbb{N}$ onto $\{a\}$ is an enumeration of $\{a\}$. But this definition does ensure that any enumerable set is intuitively countable: since any enumerable set X is going to be no larger than some subset of $\mathbb{N}$, by Enumeration 42. This is captured in the following definition:

Definition 46 ([Un]Countable). X is *countable* iff there exists a function which enumerates it; it is *denumerable* iff it is countable and infinite; it is *uncountable* if it is not countable.

Obviously, all finite sets are enumerable, since if f is a bijection showing X to be finite, f^{-1} exists and is an enumeration of X. But some infinite sets are countable too: obviously, $\mathbb{N}$ itself (there is a no bijection between $\mathbb{N}$ and some initial subset of $\mathbb{N}$, but the identity function enumerates it).

The notion of cardinality can be extended to infinite sets too (Machover, 1996: chs. 3, 6).

Definition 47 (Equinumerosity). X and Y are *equinumerous* iff there exists a bijection from X to Y.

Definition 48 (Cardinality). For each set, X – whether finite or infinite – there exists its cardinality, $|X|$, meeting the following condition: for any X and Y, $|X| = |Y|$ iff X and Y are equinumerous.

Certainly treating the natural numbers as cardinals of finite sets meets the condition in Definition 48. But Definition 48 applies to any sets; it's just that, for the infinite sets, we have no pre-theoretical grasp on what their cardinalities might be. That's okay though: we can characterise them.

Definition 49 ($\leq$). If λ and μ are cardinals, such that $|X| = \lambda$ and $|Y| = \mu$, then $\lambda \leq \mu$ iff there is an injection from X to Y.

Theorem 14
Let $|Y| = \mu$. Then $\lambda \leq \mu$ iff there exists an X such that $X \subseteq Y$ and $|X| = \lambda$.

Proof. Obvious: by Definition 49, $\lambda \leq \mu$ iff there is an injection from X to Y iff there is a bijection from X to a subset of Y. □

Theorem 15 (Schröder-Bernstein)
$\leq$ partially orders the cardinals (Machover, 1996: 38–41).

With the notion of a cardinal, we can prove that there are more sizes of set than finite sizes, because there are more cardinals than just the natural numbers, which are the cardinalities of finite sets. Since there is no bijection from any finite subset of natural numbers to the whole set of natural numbers, the cardinality of the set of natural numbers is not any finite natural number. We introduce $\aleph_0$ to name $|\mathbb{N}|$.

Theorem 16 (Cantor)
For any set X, $|X| < |\wp(X)|$.

Proof. Define a function f such that for all $x \in X$, $f(x) = \{x\}$. Since if $x \in X$, $\{x\} \subseteq X$, f is an injection from X into $\wp(X)$, and hence $|X| \leq |\wp(X)|$ by Definition 49.

To show that $|X| < |\wp(X)|$, we need to show that $|X| \neq |\wp(X)|$, i.e. (by Definition 48), that X and $\wp(X)$ are not equinumerous. Let g be any function from X to $\wp(X)$. The domain of g is the members of X, and the range are subsets of X, precisely the kinds of things members of X can be members of. But not every member of X needs to be a member of the subset of X picked out by $g(x)$. Let us define the set of such things, the $x \in X$ which aren't members of $g(x)$:

$$D = \{x \in X : x \notin g(x)\}.$$

Since D consists solely of members of X, $D \subseteq X$ and hence $D \in \wp(X)$. If g is a bijection, then there must be some $d \in X$ such that $g(d) = D$. Obviously, then $d \in D$ iff $d \in g(d)$. But by definition of D, $d \in D$ iff $d \notin g(d)$. So if g is a bijection, there is some d such that $d \in g(d)$ iff $d \notin g(d)$, and since there obviously is no such d, then g cannot be a bijection. Since g was arbitrary, *no* function from X to $\wp(X)$ is a bijection, and therefore X and $\wp(X)$ are not equinumerous. □

Cantor's theorem shows us that the powerset of a set is always a strictly larger cardinality than the set. We already know that there is one countable infinite set,

$\mathbb{N}$. Are there uncountable sets? Yes, obviously: $\wp(\mathbb{N})$: the set of all sets of natural numbers. This allows us to show that another set is uncountable: the real numbers, numbers with arbitrarily many decimal places. Indeed, we can show the following, from which it follows trivially that the set of reals is uncountable.

Theorem 17 (Uncountability of the Unit Interval)
The set of real numbers between 0 *and* 1 *inclusive,* $[0, 1]$*, is uncountable.*

Proof. (Sketch.) Each subset of $\mathbb{N}$ can be associated with an infinite binary sequence (sequence of 0s and 1s): the sequence corresponding to $X \in \wp(\mathbb{N})$ is the one which has a 1 at position n iff $n \in X$. Each infinite binary sequence corresponds to exactly one real number in $[0, 1]$. So there is a bijection from $\wp(\mathbb{N})$ to $[0, 1]$, so $|\wp(\mathbb{N})| = |[0, 1]|$, and since the former is uncountable, so is the latter. □

A final theorem linking countability and set theory, useful – perhaps – for some exercise or other.

Theorem 18 (Countable Unions)
If $\mathbf{X} = \{X_1, \ldots, X_n, \ldots\}$ *is countable, and each* X_i *is countable, then* $\bigcup \mathbf{X}$ *is countable.*

I sketch the proof. Let P be the set of positive prime numbers. This set is in one-one correspondence with $\mathbb{N}$ (for every n, there is one and only one n-th prime). Since $\mathbf{X}$ is countable, there is a one-one function from $\mathbf{X}$ into the natural numbers, so there is a one-one function f from $\mathbf{X}$ into P.

Since each X_i is countable, there is a function g_i that is a one to one correspondence between X_i and a subset of $\mathbb{N}$. Define the function $h_i(x) = f(X_i)^{g_i(x)}$. Since every integer has a unique prime decomposition – by the fundamental theorem of arithmetic (Gowers, 2008: §V.14) – if $h_i \neq h_j$ then $\mathscr{R}(h_i) \cap \mathscr{R}(h_i) = \varnothing$.

Now define: $h(x) = h_i(x)$ where $x \in X_i$ and x is not a member of any X_j where $j < i$. Clearly $\mathscr{D}(h) = \bigcup \mathbf{X}$. Obviously $\mathscr{R}(h) \subseteq \mathbb{N}$. And h is one-one, since each h_i is a function and their ranges are disjoint. So $\bigcup \mathbf{X}$ is in one-one correspondence with a subset of $\mathbb{N}$ so is countable.

Further Reading

Another presentation of the set theory needed for this book can be found in Beall and van Fraassen (2003). A presentation which goes well beyond what we need, but is philosoph-

ically nuanced, is Potter (2004). The iterative conception of a set is discussed by Boolos (1971).

Exercises

1. Prove:
   (a) If $X \subseteq Y$ and $Y \subseteq X$ then $X = Y$.
   (b) If $X \subseteq Y$ and $Y \subseteq Z$ then $X \subseteq Z$.
   (c) $X \not\subset X$.
   (d) If $X \subset Y$ then $Y \not\subset X$.
   (e) If $X \subset Y$ then $X \subseteq Y$.
2. Prove:
   (a) $X \cap X = X \cup X = X$.
   (b) $X \cap \varnothing = \varnothing$.
   (c) $X \cup \varnothing = X$.
   (d) $X \subseteq Y$ iff $X \cap Y = X$ iff $X \cup Y = Y$.
   (e) $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$.
   (f) $X \setminus (X \cap Y) = X \setminus Y$.
   (g) $X \setminus (Y \cap Z) = (X \setminus Y) \cup (X \setminus Z)$.
3. Prove that if there is a universal set Ω, then the axiom of separation entails a contradiction.
4. Let us define a *Kuratowski ordered pair* $\langle\!\langle x, y \rangle\!\rangle$ as the set $\{\{x\}, \{x, y\}\}$. Prove that
   (a) $\langle\!\langle x, y \rangle\!\rangle = \langle\!\langle u, v \rangle\!\rangle$ iff $x = u$ and $y = v$.
   (b) If $x = y$ then $\langle\!\langle x, y \rangle\!\rangle = \{\{x\}\}$.
5. (a) Prove that if X has n members, $\wp(X)$ has 2^n members.
   (b) Prove that $X \subseteq Y$ iff $\wp(X) \subseteq \wp(Y)$.
   (c) Show, by providing a counterexample, that it is not always true that $\wp(X) \cup \wp(Y) = \wp(X \cup Y)$.
6. Which of the relations expressed by the following English predicates are equivalence relations:

(a) 'x and y attend the same lectures', on the domain of Oxford students.

(b) 'x is studying the same subject as y', on the domain of Oxford students.

7. Prove the following:

(a) If R is irreflexive, then R' is reflexive.

(b) If R is symmetric, then R^{-1} is symmetric.

(c) If R is symmetric, then R' is symmetric.

(d) If R is transitive, then R^{-1} is transitive.

(e) If R is connected, then R^{-1} is connected.

(f) If R is asymmetric and non-empty, then R' is non-symmetric.

8. (a) If R is transitive, is R' transitive?

(b) If R is connected, is R' connected?

(c) If R is antisymmetric, is R' symmetric?

(d) If R is transitive and non-empty, is R' is transitive?

9. If R is defined on the empty domain, can it be reflexive? Can it be irreflexive? What about transitivity and symmetry?

10. What is wrong with the following argument that reflexivity is a consequence of symmetry and transitivity?

> If $\langle x, y\rangle \in R$, then $\langle y, x\rangle \in R$ since we assume R is symmetric. If both $\langle x, y\rangle \in R$ and $\langle y, x\rangle \in R$, then since R is transitive, $\langle x, x\rangle \in R$ – so R is reflexive. (After Partee *et al.* 1990: p. 52.)

11. (a) Can a relation be asymmetric and reflexive?

(b) Can a relation be transitive, non-symmetric and irreflexive?

(c) Can a relation be connected and irreflexive?

12. A relation R satisfies *trichotomy* iff, for all x and y, at most one of these three holds: Rxy, Ryx, or $x = y$. Is every trichotomous relation connected? Under what circumstances is a connected relation trichotomous?

13. Show that a well-ordering of D is a total ordering of D, but not *vice versa.*

14. Prove that

(a) If x is a least member under an ordering R, then it is the unique least member.

(b) The set of natural numbers $\mathbb{N}$ is well-ordered by $<$.

(c) The set of (positive and negative) integers $\mathbb{Z}$ is *not* well-ordered by $<$.

(d) There exists orderings with no maximal elements.

15. A relation R is *dense* iff whenever $\langle x, y\rangle \in R$, there exists a z such that $\langle x, z\rangle \in R$ and $\langle z, y\rangle \in R$. Prove that on the domain of the natural numbers $\mathbb{N}$, the greater-than relation $>$ is not dense; but that on the domain of the positive rationals (i.e., numbers of the form $\frac{n}{m}$ where $n, m \in \mathbb{N}$), it is dense.

16. Let $D = \{1, 2, 3, 5, 6, 10, 15, 30\}$. Let R be the relation on D defined by

$$R = \{\langle x, y\rangle : x \text{ divides } y \text{ without remainder}\}.$$

(a) Show that R is a weak partial order but not a total order.

(b) Draw the graph of R, and identify any minimal, maximal, least or greatest elements.

(c) Do the same for the set $\wp\{a, b, c\}$, using the relation $\subseteq$ on that domain.

17. (a) Show that if f^{-1} is the inverse of f, then f is also the inverse of f^{-1}.

(b) Show that f has an inverse function iff it is one-one.

(c) Give an example of a function that

i. is into but is not onto;

ii. is one-one but is not a bijection;

iii. has no inverse;

iv. is its own inverse. (A function which is its own inverse is known as an *involution*.)

(d) Under exactly what conditions is the union of two functions itself a function? (I.e., state necessary and sufficient conditions.)

(e) Give an example of a set which is

i. Countable;

ii. Denumerable;

iii. Uncountable. (Hint: use Theorem 16 and 6.(d).ii.)

18. (a) Show that if X is countable, then if $Y \subseteq X$, Y is countable.

(b) Show that if X and Y are both countable, $X \times Y$ is countable. (Hint: show that the set of ordered pairs of natural numbers is countable.)

Answers to selected exercises on page 239.

Part II

Sentential Logic: Language and Meaning

Chapter 3

The Syntax of $\mathscr{L}_1$

3.1 Strings and Quotation

The Alphabet of $\mathscr{L}_1$ The language $\mathscr{L}_1$ contains an *alphabet*, consisting of the following three types of characters:

1. *Sentence letters* $P, Q, R, P_1, Q_1, R_1, P_2, Q_2, R_2, \ldots$.
2. *Logical connectives*: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.[1]
3. *Parentheses*: (,)

No character falls into more than one category; that is, e.g., no sentence letter is also a connective. Also, no character in any category is identical to any other character in that category: e.g., '$\wedge$' is distinct from '$\neg$'.

Using this alphabet, we will now define the syntax, or grammar, of $\mathscr{L}_1$. That is, we will give rules which tell us which combinations of symbols form grammatical *sentences*. To do this, we will have to talk *about* sentence letters and sentences. Two devices of reference to strings will enable us to do this clearly.

Definition 50 (String). A *string* in a language $\mathscr{L}$ is any finite ordered sequence of characters from the alphabet of $\mathscr{L}$.

[1] The notation I use for the logical connectives is standard, but there are a couple of other standards too. One very common variant is to use '&' for '$\wedge$'. Beall and van Fraassen (2003: 25) use '~' where we use '$\neg$', '$\supset$' where we use '$\rightarrow$', and '$\equiv$' where we use '$\leftrightarrow$'.

I use lowercase Greek letters – ϕ, ψ, χ and so on – as *variables* for strings (see Appendix B).

There is a convenient way that our metalanguage – English – allows us to name strings. The convention is this. If ϕ is a string in $\mathscr{L}_1$, then the English expression consisting of left quotation mark, followed by the string ϕ, followed by the right quotation mark, is the name of ϕ (Richard, 1986: 397).[2]

Corner Quotes A problem with this convention is that we are limited in our ability to quantify into quotation, which we will want to do in specifying our syntax. Here's an example to clarify what I mean by this. Suppose we wanted to be more precise about what a sentence letter is. We might say something like this:

Sentence Letters A sentence letter is an instance of one of the following: '$\boldsymbol{P}_n$', '$\boldsymbol{Q}_n$', '$\boldsymbol{R}_n$', for $n \geqslant 0$.

But this doesn't work: for, according to our convention, '$\boldsymbol{P}_n$' is the name of the string that contains a capital italic instance of the 16th letter of the alphabet, followed by a lowercase subscripted italic instance of the 14th letter of the alphabet. The 'n' that occurs in the string is being mentioned, not used, so is not a variable that can be governed by the condition '$n \geqslant 0$'. We can get around this, by introducing a device due to Quine (1940: §6): *corner quotes* (sometimes called *quasi-quotation*).

Here's how they work. If ϕ is a string, the expression $\ulcorner\phi\urcorner$ is the quotation name of that string – i.e., it consists of the left quote mark, the string itself, then the right quote mark. (So if ϕ is a string, $\ulcorner\phi\urcorner$ is a name of the string, while 'ϕ' is the name of the variable that – temporarily – denotes the string.) Crucially, this construction allows variables to do their job: in forming $\ulcorner\phi\urcorner$, we replace the variable by whatever it happens to denote, rather than talking about the variable itself. This applies to all variables in the quasi-quotation. The quasi-quotation $\ulcorner\phi$ and $\psi\urcorner$ is the expression beginning with a left quote, followed by whatever ϕ denotes, followed by ' and ', followed by whatever ψ denotes, followed by a right quotation mark.

[2] Of course, 'ϕ' – the expression consisting of an instance of this lowercase letter of the Greek alphabet – is also being used, temporarily, to refer to that string, but it is a variable, not a name. Compare the pronoun 'him' with the name 'Antony', both of which can be used by you to refer to me, but the former only does so temporarily and under special circumstances.

Now we can see how to fix up our definition of sentence letters:

Definition 51 (Sentence Letters). A symbol is a sentence letter iff it is an instance of one of the following: $\ulcorner P_n \urcorner$, $\ulcorner Q_n \urcorner$, or $\ulcorner R_n \urcorner$, for $n \geqslant 0$. For simplicity, we write 'P' for 'P_0'.

This can all seem a bit complicated, but we need to be precise if we are going to be sure that we can prove what we need to prove about our language. Pedantry is almost always a virtue in logic.

But one also needs to know when pedantry becomes the enemy of clarity. So in what follows, I'll often drop quotes and corner quotes when there is no chance of confusion. In particular, since the object language expressions of $\mathscr{L}_1$ are not ordinary English expressions, it won't be confusing in general if I write things like: 'P_{17} is a sentence letter' rather than the strictly correct ''P_{17}' is a sentence letter'.

3.2 Sentences

Making use of corner quotes and variables over strings of $\mathscr{L}_1$, we may now specify our *syntax*. We do it by specifying which things are the (grammatical) sentences of our language.

Definition 52 (Sentences of $\mathscr{L}_1$). The *sentences* of $\mathscr{L}_1$ are those strings in the smallest collection including all strings satisfying these two conditions:

1. All sentence letters are sentences of $\mathscr{L}_1$.
2. If ϕ and ψ are sentences of $\mathscr{L}_1$, then so are:
    - $\ulcorner \neg \phi \urcorner$;
    - $\ulcorner (\phi \wedge \psi) \urcorner$;
    - $\ulcorner (\phi \vee \psi) \urcorner$;
    - $\ulcorner (\phi \rightarrow \psi) \urcorner$; and
    - $\ulcorner (\phi \leftrightarrow \psi) \urcorner$.

To say that the sentences are strings in the *smallest collection* including all strings satisfying the condition means this: every string satisfying the conditions is in the smallest collection of them, and no extra strings are in it. So, in effect, we are saying: the sentences include those strings which meet the conditions, and no other string that doesn't meet the conditions is a sentence.

Some Useful Terminology

Definition 53 (Arity). The *arity* of a connective is the number of constituent sentences the connective requires to form a grammatical sentence.

'$\neg$' is called a *unary* connective; it makes a sentence when it is supplied with one sentence. '$\wedge$', '$\vee$', etc., are called *binary* connectives, because they make a sentence when supplied with two sentences, which they connect together (hence the name). $\mathscr{L}_1$ does not involve ternary or higher arity connectives.

Definition 54 (Scope). The *scope* of an occurrence of a connective in a sentence ϕ is smallest sentence occurring as a constituent of ϕ that contains this occurrence of the connective.

So in the sentence $((P \wedge Q) \vee R)$, the scope of $\wedge$ is $(P \wedge Q)$.

Definition 55 (Main Connective). The *main connective* of a sentence is the connective (if there is any) which is in the scope of no other; alternatively, whose scope is the whole sentence.

Syntax The foregoing suffices to specify the syntax of our language $\mathscr{L}_1$. The language contains an alphabet, which includes a set of sentence letters, and a set of connectives, and some other punctuation. And it contains a *syntax*, that is, a set of rules specifying which strings constructed from the alphabet are well-formed sentences. A language contains a syntax, but to complete our specification of the language we will also need to talk about the meanings of sentences, or its *semantics*; we look at the semantics of $\mathscr{L}_1$ in chapter 4.

Abbreviatory Conventions When writing sentences of $\mathscr{L}_1$, we will often from now on drop parentheses in accordance with the following conventions (Halbach,

2010: §2.3). Note these are only conventional abbreviations: they form no part of the official syntax, but are just for our own convenience.

1. The *outermost* pair of parentheses of a sentence may be dropped; that is, we may write '$P \vee Q$' in place of the official '$(P \vee Q)$'.

2. If $\oplus$ is one of the binary connectives '$\vee$', '$\wedge$' of $\mathscr{L}_1$ (that is, '$\oplus$' is a variable over binary connectives), then we may write any sentence of the form $\ulcorner((\phi \oplus \psi) \oplus \chi)\urcorner$ as $\ulcorner(\phi \oplus \psi \oplus \chi)\urcorner$.

We may combine these conventions, dropping the outer parentheses to yield $\ulcorner\phi \oplus \psi \oplus \chi\urcorner$. Be careful: the conventions are designed to ensure that one can get back the original sentence without ambiguity. Since $P \vee Q \vee R$ is a legitimate abbreviation of $((P \vee Q) \vee R)$ in accordance with the conventions, it *cannot* be used to abbreviate $(P \vee (Q \vee R))$. That latter sentence can only be abbreviated to $P \vee (Q \vee R)$.

3.3 Proofs About Syntax

We can be quite precise about the syntax of $\mathscr{L}_1$. For an example of what we can show, consider this result:

Theorem 19 (Parenthesis Matching)
If ϕ is a sentence of $\mathscr{L}_1$, then if ϕ contains any left parenthesis '(', then it terminates with a right parenthesis ')', which is paired with the leftmost left parenthesis in ϕ (Shapiro, 2013: §2, Theorem 1).

Proof. By clause (3) of Definition 52, every sentence is built up from the sentence letters using the subclauses of clause (2) of Definition 52. Sentence letters have no parentheses (as no sentence letter is also a parenthesis). Parentheses are introduced only in subclauses involving '$\wedge$', '$\vee$', '$\rightarrow$', and '$\leftrightarrow$', and each time they are introduced as a matched set. So at any stage in the construction of a sentence, the parentheses are paired off. Moreover, each newly introduced pair of parentheses will be the leftmost and rightmost characters in a sentence. Since the only complex sentences not formed from these parenthesis-introducing rules are formed by negating existing sentences, and negation introduces a new character only at the left. So any complex sentence which contains any parenthesis terminates in a right parenthesis, which is matched to the leftmost left parenthesis. □

Now, this isn't particularly interesting on its own; it's fairly obvious from the rules for the construction of sentences. However, it's important for two reasons. First, it shows how to construct a proof in the metalanguage: a systematic argument that a certain claim – the theorem – is correct. This is a fairly informal proof of this sort, and things will get a bit more precise later in the book, but it is illustrative of the kind of thing we'll do. Second, we can use this theorem in proving further theorems. For example:

Theorem 20 (Prefix-free)
Let ϕ, ψ be nonempty strings, such that $\ulcorner \phi\psi \urcorner$ (i.e., ϕ followed by ψ) is a sentence. Then ϕ is not *a sentence (Shapiro, 2013: §2, Theorem 5).*

Proof. Either ϕ contains at least one parenthesis, or it does not.

By Theorem 19, if ϕ contains any parentheses, its leftmost parenthesis is matched to the terminal parenthesis in ψ. Since every sentence has the same number of left and right parentheses (convince yourself that this is true in the exercises), and $\ulcorner \phi\psi \urcorner$ is a sentence, and ψ is non-empty, that means ϕ must contain more left parentheses than right parentheses, and is therefore not a sentence.

So now suppose ϕ does not contain any parentheses. If it is a sentence, it must either be a sentence letter, or a negated sentence letter. If it is either of these categories, and yet $\ulcorner \phi\psi \urcorner$ is a sentence, ψ must be empty; but it's not. So it cannot be a sentence.

So, either way, ϕ is not a sentence of $\mathscr{L}_1$. □

Informally, what Theorem 20 shows is that no sentence of $\mathscr{L}_1$ is an initial substring of any other sentence. The set of grammatical sentences is thus *prefix-free*; in this way it is like the set of well-formed phone numbers (in which no phone number is the prefix of any other).[3]

3.4 Proof By Induction on Complexity

An important sort of inductive argument is *induction on the complexity of sentences.*

[3]The prefix-free nature of telephone numbers is very useful, since once the telephone system detects that a well-formed phone number has been entered, it can immediately call the number. If phone numbers were not prefix free, to call the number which is the initial part of some other number would involve entering some further information specifying that the number has been completely entered.

Definition 56 (Complexity). The complexity of a sentence of $\mathscr{L}_1$ is the sum of the arities of the connectives occurring in it. (E.g., $\neg P$ has complexity 1, $\neg\neg(P \to Q)$ has complexity 4, etc.)

The complexity of a sentence letter is 0. Since one cannot guarantee that a sentence has two constituents of equal complexity, we must use the strong principle of induction if one wants to induce on complexity. (Also because the complexity of $\phi \oplus \psi$, where $\oplus$ is some binary connective, is the complexity of ϕ, plus the complexity of ψ, plus 2.)

In effect, we already used proof by induction on complexity in the proof of Theorem 19: we showed that the atomic sentences had matching parentheses (the base case), and showed that if some constituents had matching parentheses, then the complex sentence formed by applying the further clauses of the syntax added parentheses in the appropriate way (the induction step).

Theorem 21

Each sentence consists of a string of zero or more negation symbols concatenated with either a sentence letter or a sentence with a binary connective as its main connective.

Proof. We prove this by strong induction on complexity. Assume that the theorem holds of all subsentences of a given sentence ϕ, and we'll show it to hold of the whole sentence. There are two possibilities:

1. ϕ was formed by applying the negation clause of the formation rules, so $\phi = \neg\psi$ for some ψ. By the induction hypotheses, ψ consists of a string of zero or more negations concatenated with either a sentence letter or a sentence with a binary main connective. ϕ is formed by attaching a negation sign to the front, so ϕ is a sentence with *one* or more negations concatenated with either a sentence letter or a sentence with a binary main connective – clearly this satisfies the theorem.

2. ϕ was formed by applying one of the binary connective clauses of the formation rules. It is thus a sentence with a binary connective as its main connective, prefixed by zero negation symbols, and hence the theorem holds of it. □

3.5 The Size of $\mathscr{L}_1$

How big is the language $\mathscr{L}_1$? That is: what is the size of its set of sentences $S_{\mathscr{L}_1}$? There are infinitely many finite strings constructed from the alphabet of $\mathscr{L}_1$ – we can see this because, obviously, the set of sentence letters $\mathbf{P} = \{P_n : n \in \mathbb{N}\}$ is in one-one correspondence with the set of natural numbers, and hence is countably infinite. But $\mathbf{P} \subset S_{\mathscr{L}_1}$, so $S_{\mathscr{L}_1}$ must be infinite too. The question is: is the size of $S_{\mathscr{L}_1}$ countably or uncountably infinite?

Each sentence is finitely long – we can prove this by induction on complexity, since at each complexity stage, we only form finitely long sentences from previously formed finite sentences, and there is never a stage at which an infinite sentence is formed. So each sentence of $\mathscr{L}_1$ is a finite sequence. Moreover, while our alphabet is infinite (we are treating each sentence letter as a distinct syntactically indivisible entity, at least at our level of analysis),[4] it is nevertheless countable. We can set up some standard enumeration of the alphabet: perhaps associate the parentheses with 0 and 1, the connectives with numbers 2–6, and the sentence letters with the subsequent numbers.

This means of associating numbers with items in our alphabet is in effect a *code*, which turns strings of $\mathscr{L}_1$ into corresponding sequences of natural numbers. And it turns out this entails the set of sentences of $\mathscr{L}_1$ is only countably infinite:

Lemma 22 (Key Lemma)
There is a one-one correspondence between finite sequences of natural numbers and the set of natural numbers $\mathbb{N}$.

Proof. For each l, there are countably many finite sequences of length l. For we can map any given sequence to a unique natural number, as follows: if p_k is the k-th prime number, we map the sequence $\langle n_1, \ldots, n_l \rangle$ to the product of the first l primes, each prime p_i raised to the power of n_i:

$$\mathsf{code}(\langle n_1, \ldots, n_l \rangle) = p_1^{n_1} \times \ldots \times p_l^{n_l}.$$

Since the coding function code is one-one, by the fundamental theorem of arith-

[4] We need not do this, of course: we could use the alphabet consisting of P, Q, R, and the digits $0, \ldots, 9$ to treat the sentence letters themselves as finite strings drawn from a finite alphabet. But as we'll see, this doesn't make any difference to the size of the set of sentences of $\mathscr{L}_1$.

metic (Gowers, 2008: §V.14), it has an inverse, which is from a the natural numbers *onto* the set of finite sequences of length l, which is thus an enumeration, and hence the latter set is countable.

The set of all finite sequences of natural numbers is the union of every set of finite sequences of natural numbers of fixed length. (I.e., it is the union of: the set of such sequences of length 1, the set of such sequences of length 2, ...) There are countably many such sets (because finite lengths are natural numbers). So the set of all finite sequences is a countable union of countable sets; by Theorem 18, the set of all finite sequences of natural numbers is countable. □

By the Key Lemma, and the fact that we can model strings of $\mathscr{L}_1$ as finite sequences of natural numbers, there is a one-one correspondence between arbitrary finite strings of the alphabet of $\mathscr{L}_1$ and a countable set. Since the set of sentences of $\mathscr{L}_1$ $S_{\mathscr{L}_1}$ is a strict subset of the set of finite strings of the alphabet of $\mathscr{L}_1$ the cardinality of $S_{\mathscr{L}_1}$ is no greater than countable (by Theorem 14), and since it is infinite, it must therefore be countably infinite.

Exercises

1. Correctly punctuate the following sloppy remark using corner quote notation and removing any extraneous quotation marks

   > For any numbers 'n' and 'm', '$n + m$' denotes a number, as long as + denotes the addition operator.

2. Prove that every sentence of $\mathscr{L}_1$ has the same number of left and right parentheses.
3. Prove that if ϕ is a non-atomic $\mathscr{L}_1$ sentence, then there is exactly one formation clause from Definition 52 that could have been applied to existing sentences to produce ϕ. *(Hint: use the fact that no character is identical to any other. You'll need to show that it is not the case that, e.g.,* $(\phi \vee \psi)$ *is the same sentence as* $(\chi \wedge \xi)$*.)*
4. Assuming the result of the previous exercise, show that every sentence of $\mathscr{L}_1$ is *uniquely readable* – that is, each sentence can be produced from the sentence letters in accordance with the formation clauses in exactly one way (Shapiro, 2013: §2).

Answers to selected exercises on page 240.

Chapter 4

The Semantics of $\mathscr{L}_1$

4.1 Semantics for $\mathscr{L}_1$

To complete our specification of $\mathscr{L}_1$, we need to provide some account of the intended meanings of sentences of $\mathscr{L}_1$. In natural languages each expression has a specific meaning or meanings, and each use makes a specific contribution to the utterances in which it appears. There is some non-specificity due to ambiguity, but even that is tightly circumscribed. Our language will differ from natural language in this respect. For we will not insist on any one fixed interpretation of the sentence letters of the language.[1] However, the meanings we assign to the logical connectives will resemble more closely the kinds of meanings in natural languages: the meanings of connectives will make a fixed contribution, alongside the syntax, to determining the meanings of complex sentences.

In a sense, $\mathscr{L}_1$ and other formal languages are not really devices for communication of meaning at all. They provide ways of representing structure within sentences and arguments, and it is those parts of the language which are assigned a fixed meaning which determine which aspects of structure are able to be captured by a given formal language. $\mathscr{L}_1$ gives the connectives which combine with

[1] We resist the assignment of any fixed meaning to the sentence letters in part for practical reasons. If we wish to use our logical languages to model natural language arguments, it would be tedious indeed to have to figure out which sentence letter translates the natural language sentences we are considering. (Is it P_{371}?) So we allow the meaning of the sentence letters to be fixed anew in each application.

sentence letters to create complex sentences a fixed meaning – so it is able to represent those aspects of the structure of a sentence that it possesses in virtue of the presence of those connectives. And it can be used to model the structure of natural language sentences which feature analogues of those connectives. But it is only in a particular application that the sentence letters of $\mathscr{L}_1$ are given any meaning. A temporary assignment of meanings to those components of a language without fixed meanings is known as an *interpretation* of the language.

Meaning and Truth Values One common approach to the theory of meaning, or *semantics*, for natural languages is to identify one primary aspect of meaning of a sentence with its *truth conditions*, the circumstances under which the sentence is true. So in important respects, the meaning of 'Jonquil is walking' is given by specifying that it is true under conditions when Jonquil is walking, and false under other conditions. The actual truth value of the sentence is thus part of the meaning of the sentence, because whether a sentence is true in the actual circumstances is part of its truth conditions. And the same is true for the truth value of the sentence in other, non-actual, circumstances.

$\mathscr{L}_1$ adopts a similar approach, because it assigns a truth value to each of its sentences under each interpretation. Note that the approach is similar but nevertheless distinct: natural language semantics is concerned with truth in various actual and possible circumstances, while in formal logic we are interested in truth under various interpretations and reinterpretations of the constituents of the language – this difference has particular significance when we consider validity and entailment below. Furthermore, the theory of meaning for $\mathscr{L}_1$ states that the truth value assigned exhausts the meaning of the sentence, under that interpretation – there are no further aspects of meaning.

In practice, an intepretation of $\mathscr{L}_1$ associates a *truth value* – either True or False – with every sentence letter of $\mathscr{L}_1$. Having interpreted (or re-interpreted) the sentence letters, the remainder of the semantics is constructed to be such that the truth-value of a sentence depends on its semantically relevant structure and the truth-values of its constituents. $\mathscr{L}_1$ thus has what is known as a *compositional semantics*: having assigned meanings to the most basic constituents of the language, the semantics specifies how those meanings are to be extended to every sentence in the language by showing how the semantic values (which are just

truth values, in $\mathscr{L}_1$) of a complex sentence are determined by its structure and the semantic values of its parts.

Translation When translating between natural languages, it is best (if possible) to translate a sentence in the home language by a sentence in the target language with the same truth conditions. Such a guideline cannot be applied to translation between natural and formal languages. What we can do is make sure that we pick an intepretation of the formal language such that the actual truth value of the natural language sentence is reflected in the truth value assigned to the formal language sentence letter. (And, of course, make sure you translate only those natural language sentences without internal logical structure that can be represented in your target formal language by sentence letters of that language.)

Classical Valuations In classical logic, as in life, there are two truth values: True T and False F (sometimes written 1 and 0, sometimes written $\top$ and $\bot$). Recall our discussion of Bivalence in chapter 1; it is a presumption of classical logic that there are only two truth values, and that each meaningful sentence possesses one and only one of them. $\mathscr{L}_1$ is a classical logic, and any semantics for $\mathscr{L}_1$ will support a total pattern of assignments of truth values to the sentences of the language that respects Bivalence, once we have a Bivalent interpretation of the sentence letters. Such a assignment of truth values is known as a valuation:

Definition 57 (Valuation). A *valuation* of a language is an assignment of values (of some sort of other) to the sentences of that language. A *classical valuation* is a valuation in which the possible values are the classical truth values T and F, and in which every sentence has exactly one value assigned to it.

We can express what we said about classical valuations more concisely using the language of functions: a classical valuation on some language is a total function from the set of sentences of a language into the set of truth values. We may also use it to define a new notion:

Definition 58 ($\mathscr{L}_1$-Structure). An *$\mathscr{L}_1$-structure* is a (total) function from the set of sentence letters of $\mathscr{L}_1$ into the set of classical truth values $\{T, F\}$.

An $\mathscr{L}_1$-Structure is a formal mathematical rendering of what we informally termed an interpretation of $\mathscr{L}_1$.

A valuation need not be compositional, in the sense that there need not be any rules governing which complex sentences get which truth values. But ours will be, and we can use the rules governing the construction of complex sentences to extend an $\mathscr{L}_1$ structure to a full classical valuation for $\mathscr{L}_1$.

Definition 59 ($\mathscr{L}_1$-valuation). When $\mathscr{A}$ is any $\mathscr{L}_1$-structure, $[\![\cdot]\!]_{\mathscr{A}}$ is a valuation function from the set of $\mathscr{L}_1$ sentences to the set of truth values $\{T, F\}$ iff it meets these conditions:

1. If ϕ is a sentence letter, $[\![\phi]\!]_{\mathscr{A}} = \mathscr{A}(\phi)$.
2. $[\![\neg\phi]\!]_{\mathscr{A}} = T$ if and only if (iff) $[\![\phi]\!]_{\mathscr{A}} = F$.
3. $[\![\phi \wedge \psi]\!]_{\mathscr{A}} = T$ iff $[\![\phi]\!]_{\mathscr{A}} = T$ and $[\![\psi]\!]_{\mathscr{A}} = T$.
4. $[\![\phi \vee \psi]\!]_{\mathscr{A}} = T$ iff $[\![\phi]\!]_{\mathscr{A}} = T$ or $[\![\psi]\!]_{\mathscr{A}} = T$ (or both).
5. $[\![\phi \rightarrow \psi]\!]_{\mathscr{A}} = T$ iff $[\![\phi]\!]_{\mathscr{A}} = F$ or $[\![\psi]\!]_{\mathscr{A}} = T$ (or both).
6. $[\![\phi \leftrightarrow \psi]\!]_{\mathscr{A}} = T$ iff $[\![\phi]\!]_{\mathscr{A}} = [\![\psi]\!]_{\mathscr{A}}$.

It is easy to see that for any $\mathscr{L}_1$-structure $\mathscr{A}$ the valuation function $[\![\cdot]\!]_{\mathscr{A}}$ is a classical valuation. Every sentence letter is assigned exactly one of T or F, and every complex sentence is assigned exactly one of T or F. (This depends on the result proved in the exercises to chapter 3 – see 240 – that each sentence of $\mathscr{L}_1$ has exactly one main connective, so that every sentence of $\mathscr{L}_1$ falls under one and only one of the clauses in the definition of an $\mathscr{L}_1$-valuation.)

What is also clear is that the definition of the valuation function specifies the meaning of the logical expressions of $\mathscr{L}_1$, the connectives. For the meaning of a complex sentence is determined by the semantic value assigned to its constituents, and the valuation function. The meaning of '$\wedge$' just is its functional role, the operator that makes a complex sentence from simpler constituents which is true iff both constituents are true. We'll make this precise when we talk about truth-functions as the meanings of connectives below section 4.10.

Definition 60 (Agreement of Structures). When S is a set of $\mathscr{L}_1$ sentence letters, let us say that two $\mathscr{L}_1$ structures $\mathscr{A}$ and $\mathscr{B}$ *agree* on S iff for each ϕ in S, $\mathscr{A}(\phi) = \mathscr{B}(\phi)$.

Since the value of a sentence is determined by the values of its constituents, it is obvious that if ψ is any $\mathscr{L}_1$ sentence, and if two $\mathscr{L}_1$ structures $\mathscr{A}$ and $\mathscr{B}$ agree on the sentence letters in ψ, then $[\![\psi]\!]_{\mathscr{A}} = [\![\psi]\!]_{\mathscr{B}}$.

Falsity clauses Here is another example of proof by induction on complexity of sentences. Recall the definition of a valuation. It told us under what circumstances a sentence of a given form was true in a valuation. Why didn't we also need to state when a sentence was false? Because a sentence is false iff it is not true.

Theorem 23 (Falsity is Untruth)
$[\![\phi]\!]_{\mathscr{A}} = F$ *iff* $[\![\phi]\!]_{\mathscr{A}} \neq T$.

Proof. Base case: ϕ is a sentence letter. Because $\mathscr{A}$ is a function, if $[\![\phi]\!]_{\mathscr{A}} = F$ then $[\![\phi]\!]_{\mathscr{A}} \neq T$. Because $\mathscr{A}$ is into and total, if $[\![\phi]\!]_{\mathscr{A}} \neq T$ then $[\![\phi]\!]_{\mathscr{A}} = F$.

Induction step: Suppose ϕ is a sentence, but is complex, and that the theorem holds for the constituents of ϕ. We show two illustrative cases:

1. Suppose $\phi = \neg\psi$. Then $[\![\phi]\!]_{\mathscr{A}} = F$ iff $[\![\neg\psi]\!]_{\mathscr{A}} = F$, iff $[\![\psi]\!]_{\mathscr{A}} = T$, iff $[\![\psi]\!]_{\mathscr{A}} \neq F$ iff $[\![\neg\psi]\!]_{\mathscr{A}} \neq T$ iff $[\![\phi]\!]_{\mathscr{A}} \neq T$.

2. Suppose $\phi = (\psi \vee \chi)$. Then $[\![\phi]\!]_{\mathscr{A}} = F$ iff $[\![\psi \vee \chi]\!]_{\mathscr{A}} = F$ iff $[\![\psi]\!]_{\mathscr{A}} = F$ and $[\![\chi]\!]_{\mathscr{A}} = F$ iff $[\![\psi]\!]_{\mathscr{A}} \neq T$ and $[\![\chi]\!]_{\mathscr{A}} \neq T$ iff $[\![\psi \vee \chi]\!]_{\mathscr{A}} \neq T$ iff $[\![\phi]\!]_{\mathscr{A}} \neq T$.

The cases of the other connectives are left for exercises. □

4.2 Truth Tables

One way of representing $\mathscr{L}_1$ structures – or at least, representing as much of them as matters in some particular application – is using a *truth table*. Consider any finite set of $\mathscr{L}_1$ sentences, Γ. Since each sentence in Γ is also only finitely long, there are at most finitely many sentence letters occurring in Γ. The values assigned to each of these sentence letters, in accordance with Definition 59, determines the values assigned to every sentence in Γ.

Since there are only finitely many sentence letters in Γ, each of which is assigned either *T* or *F* by our structures, it is clear there are only finitely many ways of assigning truth values to the sentence letters. (In fact, if there are n sentence letters, there are 2^n ways of assigning them truth values.) So we can write down – in principle – each of those ways of assigning truth values in a *truth table*. This table summarises the truth values of the sentences in Γ across all possible $\mathscr{L}_1$-structures.

Constructing Truth Tables For each sentence letter that appears in some sentence within Γ, inscribe a column. To be precise, inscribe those columns in the standard order: $P, Q, R, P_1, Q_1, R_1 \ldots$. For each way of assigning truth values to sentence letters, inscribe a row, by putting a truth value under the appropriate column so that every possible combination of independent assignments of classical truth values to the sentence letters is represented in some row. Now for any way of assigning truth-values to sentence letters in Γ, there is a row of the truth table representing that assignment. It is obvious that each row of the truth table corresponds to a class of structures: namely, a class of structures that all agree on the sentence letters in Γ, as per Definition 60. (Each distinct structure disagrees over some sentence letter, but the structures corresponding to a single row in a truth table for Γ only disagree over sentence letters not occuring in Γ – since they are not relevant to determining the truth value of any sentence in Γ, such distinctions do not matter for drawing up the truth table.) Now add a column for each sentence in Γ, and for each row, inscribe under the sentence the value which is assigned to that sentence by any structure corresponding to that row. (Since all such structures agree on the sentence letters in the sentence, they will all agree on what they assign to the sentence too, so it doesn't matter which one we pick.)

So, for instance, suppose Γ is this interesting set of $\mathscr{L}_1$ sentences:

$$\{\neg P, (P \wedge Q), (P \vee Q), (P \rightarrow Q), (P \leftrightarrow Q)\}\,.$$

The truth table for this set of sentences is pictured in Table 4.1.

P	Q	$\neg P$	$(P \wedge Q)$	$(P \vee Q)$	$(P \rightarrow Q)$	$(P \leftrightarrow Q)$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Table 4.1: Truth Table for the Standard Connectives

4.3 Satisfaction, Entailment, and other Semantic Notions

Many conceptions of logic have it that logic is fundamentally about *consequence*: what it is for some sentences to follow from one another, in virtue of the logical form of the sentences involved (Beall and Restall, 2013). But what is consequence? We may characterise it semantically as follows: some sentences of $\mathscr{L}_1$ have another sentence as a consequence if the former sentences *entail* the latter sentence. And we may characterise entailment, and a number of other semantic relations between sentences, using the notions we've already introduced.

Definition 61 (Satisfaction). Suppose Γ is any (possibly empty, possibly finite, possibly infinite) set of sentences of $\mathscr{L}_1$, and $\mathscr{A}$ is a $\mathscr{L}_1$-structure, such that $[\![\gamma]\!]_{\mathscr{A}} = T$ for every sentence $\gamma \in \Gamma$. In that case, $\mathscr{A}$ *satisfies* Γ, or $\mathscr{A}$ is a *model* of Γ.

Definition 62 (Entailment). A set of sentences Γ *(semantically) entails* a sentence ϕ iff every $\mathscr{L}_1$-structure which satisfies Γ also satisfies ϕ. Notation: $\Gamma \vDash \phi$.

Definition 63 (Tautology). ϕ is a *tautology* iff every $\mathscr{L}_1$ structure satisfies $\{\phi\}$. A tautology is sometimes called a *logical truth.*

Theorem 24
If ϕ is a tautology, then for any set of sentences Γ, $\Gamma \vDash \phi$ – even if Γ is the empty set, which contains no sentences at all.

This theorem explains the fact that we often write '$\vDash \phi$' to mean that ϕ is a tautology.

If no $\mathscr{L}_1$-structure satisfies Γ, Γ is *unsatisfiable*, or *semantically inconsistent*, which we write $\Gamma \vDash$.[2] Accordingly, a set of sentences is *semantically consistent* iff it is satisfiable. If $\Gamma = \{\phi\}$ and $\Gamma \vDash$ then ϕ is a *contradiction.*

Theorem 25
ϕ is a tautology iff $\neg\phi$ is a contradiction.

Proof. ϕ is a tautology iff $[\![\phi]\!]_{\mathscr{A}} = T$ in every $\mathscr{L}_1$ structure $\mathscr{A}$. By Definition 59, this is the case iff $[\![\neg\phi]\!]_{\mathscr{A}} = F$ in every $\mathscr{L}_1$ structure $\mathscr{A}$; iff no $\mathscr{L}_1$ structure satisfies $\{\neg\phi\}$, i.e., $\neg\phi$ is a contradiction. □

Theorem 26 (Entailment and Unsatisfiability)
$\Gamma \vDash \phi$ *iff* $\Gamma, \neg\phi \vDash$.

Proof. $\Gamma \vDash \phi$ iff every $\mathscr{L}_1$-structure which makes all of Γ true makes ϕ true; iff every $\mathscr{L}_1$-structure which makes all of Γ true makes $\neg\phi$ false; iff there is no $\mathscr{L}_1$-structure which makes all of Γ true along with $\neg\phi$; iff $\Gamma \cup \{\neg\phi\}$ is unsatisfiable.□

This elementary theorem is nevertheless rather useful. One thing it shows is that *reductio* reasoning is good in $\mathscr{L}_1$: for if we have an unsatisfiable set, we can pick any member of that set, negate it, and conclude that the remaining members entail it. (Remember that $\Gamma, \phi \vDash$ just means that the set $\Gamma \cup \{\phi\}$ is unsatisfiable – order does not matter.) As we have already seen, it is often easier in practice to prove $\Gamma \cup \{\neg\phi\}$ unsatisfiable than to come up with a constructive argument from Γ to ϕ.

4.4 Consequences and Theories

With the notion of entailment in hand, we can explore the content of a set of sentences by seeing what follows from it. That way we don't focus on the particular form of the sentences involved, but on the power of

Definition 64 (Consequences). The set of *consequences* of a set of $\mathscr{L}_1$ sentences Γ, which we write $Cn(\Gamma)$, is defined as the set including every sentence entailed by Γ: $Cn(\Gamma) = \{\phi : \Gamma \vDash \phi\}$.

[2] We use this notation to make the notation for unsatisfiability mirror the notation for tautologousness – don't worry about trying to understand unsatisfiability as entailment of the empty set or anything like that – just remember that '$\Gamma \vDash$' is shorthand for 'Γ is unsatisfiable'.

Some preliminary results about the consequences of a set of sentences.

Lemma 27
$\Gamma \subseteq Cn(\Gamma)$.

Proof. Left for exercise. □

Lemma 28
$Cn(Cn(\Gamma)) = Cn(\Gamma)$.

Proof. If $Cn(Cn(\Gamma)) \neq Cn(\Gamma)$, then there must be some $\phi \in Cn(Cn(\Gamma))$ which is not a consequence of Γ (by Lemma 27). So there is a structure $\mathscr{A}$ in which ϕ is false but Γ is satisfied. Since Γ is satisfied in $\mathscr{A}$, so is $Cn(\Gamma)$. But then so is ϕ; contradiction. So there is no such ϕ – hence $Cn(Cn(\Gamma) = Cn(\Gamma)$. □

Lemma 29
$Cn(\Gamma \cup \Delta) = Cn(Cn(\Gamma) \cup Cn(\Delta))$.

Proof. Left for exercise. □

A theory, informally, is some systematic body of ideas. A theory in $\mathscr{L}_1$ is a body of $\mathscr{L}_1$ sentences that is closed under entailment, that is, every logical consequence of the theory is part of the theory.

Definition 65 (Theory). A set of $\mathscr{L}_1$ sentences Γ is a *theory* iff $\Gamma = Cn(\Gamma)$.

Definition 66 (Negation Complete). A set of $\mathscr{L}_1$ sentences Γ is *negation-complete* iff for every $\mathscr{L}_1$ sentence ϕ, either $\phi \in \Gamma$ or $\neg\phi \in \Gamma$.

Theorem 30 (Identity of Theories)
If Θ_1 and Θ_2 are theories, such that

1. *If $\Theta_1 \vDash \phi$, then $\Theta_2 \vDash \phi$;*
2. *Θ_1 is negation-complete; and*
3. *Θ_2 is satisfiable,*

then $\Theta_1 = \Theta_2$.

Proof. Condition (1) entails that $Cn(\Theta_1) \subseteq Cn(\Theta_2)$. Condition (2) entails that if $\phi \notin \Theta_1$, then $\Theta_1 \cup \{\phi\}$ is unsatisfiable. Because Θ_1 is a theory, $\Theta_1 = Cn(\Theta_1)$. So if $\phi \notin Cn(\Theta_1)$, then $Cn(\Theta_1) \cup \{\phi\}$ is unsatisfiable. So if there is any sentence in $Cn(\Theta_2)$ that is not in $Cn(\Theta_1)$, then $Cn(\Theta_2)$ is unsatisfiable, and therefore Θ_2 is unsatisfiable. Condition (3) says that Θ_2 is satisfiable, so there is no member of Θ_2 which is not a member of Θ_1, i.e., $\Theta_1 = \Theta_2$. □

4.5 Entailment, Validity and Necessity

An *argument* involves a set of sentences Γ, its *premises*, and a single sentences ϕ as its conclusion.

Definition 67 (Properties of arguments). An argument from premises Γ to conclusion ϕ is *(logically) valid* iff Γ entails ϕ. A valid argument is *sound (under a given interpretation)* iff each of its premises is actually true under the interpretation.

Necessary Truth Preservation? Sometimes one sees a purported definition of validity (or entailment, or [logical] consequence) along these lines: that an argument is valid iff it is *impossible* for its premises all to be true while its conclusion is false (Henle, Garfield, and Tymoczko, 2011: 19). That is, if there is no possible situation in which the premises are true and the conclusion false. While there are some similarities between $\mathscr{L}_1$-structures and possible situations, this sort of definition of validity should be resisted. We can think of a possible situation as specified by a description of a *way things could have been*. Importantly, the description will be in a certain actual language, which we need to hold fixed in order for it to do its descriptive job. But an $\mathscr{L}_1$-structure is, in some ways, precisely the opposite of this. It is a way of interpreting the sentence letters of $\mathscr{L}_1$, while holding fixed the world at which those reinterpreted sentences are to be evaluated. In many cases it makes no difference whether we consider a sentence P to be evaluated at a possible situation in which what P says obtains, or to be evaluated at actuality under an interpretation which makes it true. But there are cases in which it matters. Consider this argument:

> Sylvester is a child;
> Therefore, Sylvester is not an adult.

The premise *necessitates* the conclusion, for there is no possible world in which a child is an adult. But this argument isn't logically valid; for under a reinterpretation of the non-logical vocabulary 'adult' on which it means 'child', the premise is actually true and the (reinterpreted) conclusion actually false. The premise guarantees the truth of the conclusion, in part because of what the non-logical expressions 'adult' and 'child' actually mean. But in logic, we are interested in arguments where the premises guarantee the truth of the conclusion in virtue only of the meanings of the logical expressions involved (Tarski, 1936).

Formality Why does logic concern itself with what follows from a sentence under every possible reintepretation of the non-logical expressions (involved (i.e., those without a fixed meaning - the sentence letters, in the case of $\mathscr{L}_1$)? This is because logic is primarily concerned with that special case of consequence in which the conclusion follows from the premises in virtue of the syntactic form of the sentences involved. To focus on syntactic structure, we ought to neglect the actual meaning of the basic constituents of the language, and focus only on how those constituents are deployed into a complex syntactic structure. It is in a sense like replacing those basic constituents with nonsense – but once that replacement is made, we can see that the resulting argument still might be good in virtue of its structure. 'Fleebles bork and greebles gork' entails 'Fleebles bork', no matter what those constituent expressions happen to mean, just as long as 'and' has *its* actual meaning. What the above example shows is that there are arguments where the premises necessitate the conclusion, but not in virtue of their logical form – at least, not in virtue of those aspects of logical form able to be represented in $\mathscr{L}_1$. Perhaps in some logic, we could analyse 'child' as 'non-adult', and in such a logic the argument above would be logically valid. This shows us something else interesting: that logical form depends on which expressions one takes to be logical expressions, and that may vary between different logical languages.

These remarks also reflect on the notion of a tautology. Our conception is this: a tautology is true in virtue of its logical structure, no matter what the meaning of its non-logical constituents. This suffices for necessity, but the converse is not true. There are necessary truths which are not logical truths – for example, 'Adults are not children' is necessary but not a tautology of $\mathscr{L}_1$.

4.6 Meaning, Possibility, and Time

The picture we have of meaning in our language $\mathscr{L}_1$ is this. A sentence letter has as its meaning, relative to an $\mathscr{L}_1$ structure, a truth value. If a possible scenario is which has a coherent description in a given language, then we can define a $\mathscr{L}_1$-*logically possible scenario* as one described by a satisfiable set of $\mathscr{L}_1$ sentences. I will say something briefly about the philosophical issues about meaning involved in this picture.

There are several respects in which this language differs from natural language. We have just noted, in effect, that being true in every possible scenario relative to $\mathscr{L}_1$ is to be distinguished from being genuinely necessary. It is not really possible to make each sentence in this set jointly true: {'Sylvester is a child', 'Sylvester is an adult'}. But the only possible translations of these sentences into $\mathscr{L}_1$ turn that set into a satisfiable one, so the scenario is logically possible. (This probably shows that the terminology of 'logical possibility' is misleading, since it is not a type of possibility at all – perhaps 'logically coherent' or 'formally coherent' are preferable.) On the other hand, it is to be hoped that any genuinely possible scenario is also logically coherent.[3]

The treatment of possibility for $\mathscr{L}_1$ sentences is not as controversial as the treatment of time. For many natural language sentences change their truth value while keeping their meaning fixed – so truth value isn't even part of the meaning (though it might be determined by the meaning in conjunction with the circumstances the sentence is taken to be describing). So the English sentence 'It's raining in Munich' is true today, but false tomorrow, even though it seems to mean the same thing on both days. Three proposals might enable us to bring the treatment of the logical language and natural language closer on this issue.

- First, you might consider the logical language adequate to handle only that fragment of natural language which concerns sentences which have a constant truth value over time.
- Second, you might think that $\mathscr{L}_1$ sentences concern just what is *presently true*; if you wanted to handle claims about what *was true* (which we handle

[3]Even this is complicated by the fact that different logical languages might be able to express different things, as we'll see later in this book, and so there might not be any single notion of logical coherence that is a feature of any genuine possibility.

with past tense sentences in natural language), or what *will be true* (natural language future tense and related constructions), then you would need to extend the language of $\mathscr{L}_1$ to include something like tense. This is the subject of what is called, naturally enough, *tense logic*. The basic idea is to treat a structure for a sentential tense logic as a sequence of $\mathscr{L}_1$ structures, with each structure in the sequence telling us what sentence letters would be true if that structure corresponded to the present moment. So if 'It is raining in Munich' is true at time t_1 then false at time t_2, we could model that by translating the sentence by the sentence letter $\boldsymbol{P}$, letting t_1 correspond to some $\mathscr{L}_1$ structure in which $\boldsymbol{P}$ is assigned $\boldsymbol{T}$, and let t_2 correspond to some $\mathscr{L}_1$ structure in which $\boldsymbol{P}$ is assigned $\boldsymbol{F}$. A whole possible world, on this view, is then modelled by the sequence of structures – because a possible situation, to accomodate tense as an important feature of reality, must include other times in addition to the present moment.

- Alternatively, we could take it that natural language tense is eliminable in some way, so that in fact the meanings of natural language sentences have constant truth value. Surprisingly enough, this is a fairly orthodox view in natural language semantics (King, 2003; Partee, 1973). The idea is that each utterance of a sentence like 'It's raining in Munich' expresses a proposition like *It is raining in Munich on such-and-such date* – and those propositions are permanently true if true at all, because they are about a specific time. If this is right, then there is no problem assigning as the meaning of a sentence a truth value relative to a possible scenario, since the meanings of natural language sentences determine propositions of constant truth value. To do full justice to this approach, however, involves delicate issues about both natural language and about the 'untensed' nature of reality.

For practical reasons, I'll simply adopt the first proposal, and ignore tense in discussions of $\mathscr{L}_1$ (and $\mathscr{L}_2$). As we'll see when we move to talking about $\mathscr{L}_2$ later in this book, there are already natural language constructions which cannot be handled by $\mathscr{L}_1$, so there is precedent for simply taking the language of sentential logic to be able to adequately translate only some fragment of natural language.

4.7 Entailment and the Connectives

Entailment is a relation between a set of sentences and a single sentence; and a claim like $\{P, Q\} \vDash P \vee Q$ is a claim about the sentences of $\mathscr{L}_1$– it is not a statement in $\mathscr{L}_1$. But there are interesting relationships between some statements about entailment (and other semantic notions, like satisfaction), and some sentences of $\mathscr{L}_1$. We see these relationships manifest in how certain statments about entailment license further entailments between sentences involving our logical connectives in distinctive ways. For example:

Theorem 31 (Conjunction and Satisfaction)
Some sentences are satisfied in a structure iff their conjunction is satisfied in that structure. That is, if Γ *is some set of sentences* $\{\gamma_1, \ldots, \gamma_n\}$*, then for every* $\mathscr{L}_1$ *structure* $\mathscr{A}$*,* $\mathscr{A}$ *satisfies* Γ *iff* $[\![(\gamma_1 \wedge \ldots \wedge \gamma_n)]\!]_{\mathscr{A}} = \boldsymbol{T}$.

Definition 68 (Equivalence). ϕ and ψ are *logically equivalent* iff $\phi \vDash \psi$ and $\psi \vDash \phi$.

Theorem 32 (Equivalence and Biconditional)
ϕ *and* ψ *are logically equivalent iff* $\vDash \phi \leftrightarrow \psi$.

Proof. $\phi \vDash \psi$ iff every structure in which ϕ has the value $\boldsymbol{T}$ is one in which ψ also has the value $\boldsymbol{T}$, and *vice versa*. That is, iff in every structure $\mathscr{A}$, $[\![\phi]\!]_{\mathscr{A}} = [\![\psi]\!]_{\mathscr{A}}$. That holds in turn, by Definition 59, iff $[\![\phi \leftrightarrow \psi]\!]_{\mathscr{A}} = \boldsymbol{T}$ in every structure, and $\phi \leftrightarrow \psi$ is a tautology. □

Deduction Theorem Let the notation '$\Gamma, \phi \vDash \psi$' abbreviate $\Gamma \cup \{\phi\} \vDash \psi$.

Theorem 33 (Deduction)
$\Gamma, \phi \vDash \psi$ *iff* $\Gamma \vDash (\phi \rightarrow \psi)$.

Proof. $\Gamma, \phi \vDash \psi$ iff every $\mathscr{L}_1$-structure which satisfies $\Gamma \cup \{\phi\}$ also satisfies $\{\psi\}$. That holds iff there is no structure $\mathscr{A}$ in which Γ is satisfied and in which $[\![\phi]\!]_{\mathscr{A}} = \boldsymbol{T}$ and $[\![\psi]\!]_{\mathscr{A}} = \boldsymbol{F}$. By Definition 59, there is no structure $\mathscr{A}$ in which Γ is satisfied while $[\![\phi \rightarrow \psi]\!]_{\mathscr{A}} = \boldsymbol{F}$, i.e., $\Gamma \vDash \phi \rightarrow \psi$. □

Repeated applications of the deduction theorem permits us to say that an argument $\gamma_1, \ldots, \gamma_n \vDash \phi$ is valid iff $\vDash (\gamma_1 \rightarrow (\gamma_2 \rightarrow \ldots(\gamma_n \rightarrow \phi)\ldots))$ – i.e., that an argument with finitely many premises is valid iff the corresponding nested conditional is a tautology.

Corollary 34
An argument from premises Γ *to conclusion* ϕ *is valid iff the conditional with the conjunction of the premises in* Γ *as antecedent, and* ϕ *as consequent, is a tautology. That is* $\Gamma \vDash \phi$ *iff* $\vDash (\gamma_1 \wedge \ldots \wedge \gamma_n) \rightarrow \phi$.

Proof. Obvious from Theorem 31 and Theorem 33. □

The deduction theorem is the source of considerable confusion on the part of first-year logic students and others, for it seems to suggest a correspondence between entailment and the conditional. It does not: the correspondence is between entailment and a *tautologous* conditional. A conditional is true just in case – in fact – if the antecedent is true, then the consequent is also true. A tautologous conditional is one that is true on grounds of logic alone, and it is only this sort of conditional that expresses in $\mathscr{L}_1$ something analogous to entailment between sentences of $\mathscr{L}_1$. The conditional 'If we return our rental car late, we will be charged' is true, but not tautologously true – compare the genuine tautology 'If we return our rental car and we return it late, then we return our rental car', which cannot help but be true in virtue of its form ($P \wedge Q \vDash P$). We return to the topic of conditionals in chapter 15.

4.8 Structural Rules

The expression of entailment '$\Gamma \vDash \phi$' is known as a *sequent*. This sequent is correct if Γ does entail ϕ. There are a set of rules, each of which can be justified from the definitions above, that governs correctness-preserving transformations of one sequent into another. These rules are known as *structural rules*. Theorems justifying three standard structural rules follow (where Γ, Δ are sets of premises):

Theorem 35 (Permutation)
$\Gamma, \psi, \chi, \Delta \vDash \phi$ *iff* $\Gamma, \chi, \psi, \Delta \vDash \phi$ *(premise order doesn't matter).*

Theorem 36 (Contraction)
$\Gamma, \psi, \psi, \Delta \vDash \phi$ *iff* $\Gamma, \psi, \Delta \vDash \phi$ *(duplicate premises don't matter).*

Theorem 37 (Weakening)
If $\Gamma \vDash \phi$, *then* $\Gamma, \psi \vDash \phi$.

Proof. Note that every structure which satisfies all of Γ and also $\{\psi\}$ also satisfies Γ; as every structure satisfying Γ satisfies ϕ, every structure satisfying Γ, ψ satisfies ϕ. □

The structural rules are so-called because they don't depend on the meaning of any connectives, but only on the definition of entailment and the underlying set theory governing the behaviour of sets of premises. Interesting non-classical logics can arise when one varies the structural rules, which may even be done while retaining classical valuations (Restall, 2000) – of course one must vary the set-theoretic conception of entailment as a relation between sets of sentences and a sentence to vary the structural rules, in light of the theorems above.[4]

Cut, Transitivity and Contraposition

Theorem 38 (Cut)
If $\Gamma, \psi \vDash \phi$ *and* $\Gamma \vDash \psi$*, then* $\Gamma \vDash \phi$*.*

Proof. Assume that $\Gamma, \psi \vDash \phi$. If every structure in which Γ is satisfied is one in which $\{\psi\}$ is satisfied, then it is clear that the set of structures which make Γ, ψ true is just the set of structures which makes Γ true. Given that all the former satisfy ϕ, so must all the latter: $\Gamma \vDash \phi$. □

Theorem 39 (Transitivity)
If $\Gamma \vDash \psi$ *and* $\psi \vDash \phi$*, then* $\Gamma \vDash \phi$*.*

Theorem 40 (Contraposition)
$\phi \vDash \psi$ *iff* $\neg\psi \vDash \neg\phi$*.*

[4]For example, a *multiset* (sometimes known as a *bag*) is a generalisation of a set where the number of times an element occurs is significant (though order still doesn't matter). So the multisets $\wr a, a, b \wr$ and $\wr a, b \wr$ differ from one another, though $\{a, a, b\} = \{a, b\}$. If we think of entailment as a relation between multisets of premises and a conclusion, then we are unlikely to want to accept contraction: maybe the resources that the multiset $\wr P, P \wr$ provides are strictly more than those provided by $\wr P \wr$. The status of these *substructural* logical formalisms as logic – in the sense of having something to do with entailment and validity – is problematic, but their formalism has applications, often in computer science. For example, logics that do not permit contraction (*linear logics*) are often used to model computational processes where keeping track of resource use is important for implementation.

4.9 Substitution

Formality of Logic The notions of logical validity and entailment in $\mathscr{L}_1$ are *formal*: it is in virtue of the form of the premises that they entail the conclusion of a valid argument. But what does it mean to say that logic is formal? Here's one thing it could mean: take a sentence of the language, and replace some of its constituents with others of the same category, while preserving the logical structure of the sentence with respect to some particular logic. These two sentences have the same form, with respect to that logic. If we can prove some sort of equivalence between the two sentences, we will have shown that what matters in the language is form, rather than the particular constituents of a given sentence.

This is something you will have relied on implicitly in earlier logic courses: how could it be that for example P and $P \to Q$ entail Q, yet R and $R \to P_1$ fail to entail P_1? But we should prove it explicitly, to make sure that our implicit assumptions aren't leading us astray.

In the case of $\mathscr{L}_1$, the substituable constituents are sentences (including sentence letters), and the logical structure is given by the logical connectives.

Definition 69 (Constituent sentence). A *constituent sentence* of ϕ is any well-formed $\mathscr{L}_1$ sentence which occurs within ϕ as a substring. If one decomposes the sentence successively in accordance with the syntactic rules, every constituent sentence appears at some stage.

We begin by considering the substitution of sentences for sentence letters.

Definition 70 (Uniform Substitution). If Γ is a set of sentences, θ a sentence, and s a sentence letter, then write $\Gamma[\theta/s]$ for the set of sentences that results by *uniformly substituting* θ for every occurrence of constituent sentence s in every sentence in Γ. (If s doesn't occur in Γ, then $\Gamma[\theta/s] = \Gamma$.) Likewise, let $\phi[\theta/s]$ be the sentence that results from replacing every occurrence of s in ϕ by θ.

Definition 71 (Substitution Instance). If $\phi = \psi[\theta/\chi]$, for some θ, χ, then ϕ is called a *substitution instance* of ψ.

We now show a useful lemma.

Lemma 41 (Substitution)
Suppose ϕ and θ are sentences, and s a sentence letter. For any structure $\mathscr{A}$, define a structure for every sentence letter α:

$$\mathscr{A}^\star(\alpha) = \begin{cases} \mathscr{A}(\alpha) & \text{iff } \alpha \neq s \\ [\![\theta]\!]_{\mathscr{A}} & \text{iff } \alpha = s. \end{cases}$$

Then $[\![\phi]\!]_{\mathscr{A}^\star} = [\![\phi[\theta/s]]\!]_{\mathscr{A}}$.

Proof. The new structure $\mathscr{A}^\star$ is just like $\mathscr{A}$, with the possible exception that it assigns to s the value that θ has in $\mathscr{A}$.
Base case: If ϕ is a sentence letter, then the result follows by construction of $\mathscr{A}^\star$.
Induction step: If ϕ is complex, and the theorem holds for less complex claims, then the result follows. I show one case: where ϕ is a conjunction of two simpler constituents. The induction hypothesis is that $[\![\phi_i]\!]_{\mathscr{A}^\star} = [\![\phi_i[\theta/s]]\!]_{\mathscr{A}}$ for each ϕ_i which is a constituent of ϕ.

By the semantic rules for the valuation function, and the induction hypothesis,

$$\begin{aligned} [\![\phi]\!]_{\mathscr{A}^\star} = T \text{ iff } & [\![(\phi_1 \wedge \phi_2)]\!]_{\mathscr{A}^\star} = T \\ \text{iff } & [\![\phi_1]\!]_{\mathscr{A}^\star} = [\![\phi_2]\!]_{\mathscr{A}^\star} = T \\ \text{iff } & [\![\phi_1[\theta/s]]\!]_{\mathscr{A}} = [\![\phi_2[\theta/s]]\!]_{\mathscr{A}} = T \\ \text{iff } & [\![\phi_1[\theta/s] \wedge \phi_2[\theta/s]]\!]_{\mathscr{A}} = T \\ \text{iff } & [\![(\phi_1 \wedge \phi_2)[\theta/s]]\!]_{\mathscr{A}} = T. \end{aligned}$$

The desired result follows: $[\![\phi]\!]_{\mathscr{A}^\star} = [\![\phi[\theta/s]]\!]_{\mathscr{A}}$. *Mutatis mutandis* for the other connectives. □

With this lemma in hand, we can establish further results about substitution.

Theorem 42 (Substitution of Material Equivalents)
If $[\![\theta]\!]_{\mathscr{A}} = \mathscr{A}(s)$, then $[\![\phi]\!]_{\mathscr{A}} = [\![\phi[\theta/s]]\!]_{\mathscr{A}}$.

Proof. If $[\![\theta]\!]_{\mathscr{A}} = \mathscr{A}(s)$, then $\mathscr{A} = \mathscr{A}^\star$; applying Theorem 41, the result follows immediately. □

Theorem 43 (Substitution of Sentence Letters)
If Γ entails ϕ, then so too $\Gamma[\theta/s] \vDash \phi[\theta/s]$, for any θ, s.

Proof. Now suppose for *reductio* that $\Gamma \vDash \phi$, but $\Gamma[\theta/s] \nvDash \phi[\theta/s]$. In that case, there is a structure $\mathscr{B}$ such that for each $\gamma \in \Gamma$, $[\![\gamma[\theta/s]]\!]_{\mathscr{B}} = T$, but $[\![\phi[\theta/s]]\!]_{\mathscr{B}} = F$.

By the Substitution theorem 41, there is therefore a structure $\mathscr{B}^\star$ such that for each $\gamma \in \Gamma$, $[\![\gamma]\!]_{\mathscr{B}^\star} = T$ and $[\![\phi]\!]_{\mathscr{B}^\star} = F$. But in that case, $\Gamma \nvDash \phi$ after all, since there is an $\mathscr{L}_1$ structure where the premises are true and the conclusion false, contradicting our *reductio* supposition, which must have been wrong. □

Generalised Substitition and Formal Logic A generalisation of Theorem 43 is also provable. The notation '$\phi[\beta/\alpha][\gamma/\beta]$' is to represent the uniform substitution, first, of β for α throughout ϕ, and *then* the substitution of γ for β throughout the resulting sentence. It is clear that $\phi[\beta/\alpha][\gamma/\beta] = \phi[\gamma/\alpha]$, as long as β didn't already occur as a constituent sentence of ϕ.

Theorem 44 (General Substitution)
When Γ *is finite, if* $\Gamma \vDash \phi$*, then also* $\Gamma[\theta/\chi] \vDash \phi[\theta/\chi]$*, for any* θ, χ*.*

Proof. Suppose for *reductio* that $\Gamma \vDash \phi$, but there is a sentence letter s *not occurring in* Γ *or* ϕ such that $\Gamma[s/\chi] \nvDash \phi[s/\chi]$. There must be a structure $\mathscr{B}^\star$ such that for each $\gamma \in \Gamma$, $[\![\gamma[s/\chi]]\!]_{\mathscr{B}^\star} = T$, but $[\![\phi[s/\chi]]\!]_{\mathscr{B}^\star} = F$. By theorem 41, there is therefore a structure $\mathscr{B}$ such that for each $\gamma \in \Gamma$, $[\![\gamma]\!]_{\mathscr{B}} = T$ and $[\![\phi]\!]_{\mathscr{B}} = F$.[5] But in that case, $\Gamma \nvDash \phi$ after all, contradiction: so our supposition must have been wrong. So in fact for any sentence letter s not occurring in Γ or ϕ, when $\Gamma \vDash \phi$, $\Gamma[s/\chi] \vDash \phi[s/\chi]$. But since there are infinitely many sentence letters and each of the finitely many members of Γ and ϕ contains only finitely many sentence letters, we may always find such a 'new' s.

Now we may apply Theorem 43 to show that $\Gamma[s/\chi][\theta/s] \vDash \phi[s/\chi][\theta/s]$. But this latter sequent just is $\Gamma[\theta/\chi] \vDash \phi[\theta/\chi]$. □

Theorem 44 gives us another route to the formality of logic.[6] We might propose that ϕ shares a logical form with ψ if each is a substitution instance of the other.[7]

[5] This is because for any sentence γ in which s does not occur, $\gamma = \gamma[s/\chi][\chi/s]$.

[6] The reasoning is actually a little subtle: what happens when Γ is infinite? Then we cannot be assured that there is some new sentence letter occurring in neither ϕ nor Γ. The Compactness Theorem (Theorem 60), which we will prove in chapter 5, assures us that: $\Gamma \vDash \phi$ iff there is some finite set $\Gamma^- \subseteq \Gamma$ such that $\Gamma^- \vDash \phi$. We can then apply Theorem 44 to Γ^-. But don't worry about this wrinkle until then; we rarely come across arguments with infinitely many premises in everyday life....

[7] So $(P \vee Q)$ shares a logical form with $((R \wedge \neg R) \vee Q)$, because $(P \vee Q)[(R \wedge \neg R)/P] = ((R \wedge \neg R) \vee Q)$, and $((R \wedge \neg R) \vee Q)[P/(R \wedge \neg R] = (P \vee Q)$. Note however that even though $(P \vee Q)[Q/P] = (Q \vee Q)$, $(Q \vee Q)$

Any valid argument, when we uniformly and systematically substitute one constituent for another throughout every sentence involved in premises and conclusion, will transform into another argument of the same logical form; and that second argument will also be valid. Since $P, P \to Q \vDash Q$, accordingly so too $P, (P \to \neg P) \vDash \neg P$, which is a substitution instance of the former argument. But as this example shows, while we retain validity, we need not retain *soundness*. In this case no matter what interpretation we give to P, the premises of the post-substitition argument are inconsistent with each other, so we know that they cannot all be true under any interpretation. But we could have interpreted P and Q in the original argument so that they came out simultaneously true, so the argument could have been sound while its substitution instance could not. Soundness is not a matter of form, but the particular interpretation of a sentence of a given form.

Equivalence Revisited There is one case where soundness under an interpretation is *guaranteed* to be preserved: if what is substituted for one another are logically equivalent sentences.

Theorem 45 (Equivalents)
If $\Gamma \vDash \phi$ is sound under an interpretation, then so is any sequent $\Gamma[\theta/\chi] \vDash \phi[\theta/\chi]$ where θ and χ are logically equivalent, under the same interpretation.

Proof. Left for exercise. □

Theorem 46 (Equivalence)
Suppose ϕ is a constituent sentence of χ, and $\chi' = \chi[\psi/\phi]$. Then $\phi \leftrightarrow \psi \vDash \chi \leftrightarrow \chi'$.

Proof. Base case: Suppose $\chi = \phi$. Then $\chi' = \psi$, and obviously $\phi \leftrightarrow \psi \vDash \phi \leftrightarrow \psi$.

Induction step: Suppose χ is complex, e.g., $\chi = \neg\gamma$, and the induction hypothesis holds for simpler constituents: $\phi \leftrightarrow \psi \vDash \gamma \leftrightarrow \gamma'$. For any structure, $[\![\alpha \leftrightarrow \beta]\!]_{\mathscr{A}} = [\![\neg\alpha \leftrightarrow \neg\beta]\!]_{\mathscr{A}}$, so $\phi \leftrightarrow \psi \vDash \neg\gamma \leftrightarrow \neg(\gamma')$. But since $\neg\gamma = \chi$, and $\neg(\gamma') = (\neg\gamma)' = \chi'$, the result follows. (The other cases are left for exercises.) □

---

does not share a logical form with $(P \vee Q)$, since we cannot uniformly substitute for Q in $(Q \vee Q)$ and obtain $(P \vee Q)$.

		c
T	T	T
T	F	F
F	T	T
F	F	T

Table 4.2: A truth-function table for **c**

4.10 Truth-functions

Definition 72 (Truth Function). A *truth function* f_n is any total function from ordered sequences of n truth values into the set of truth values $\{T, F\}$.

Since an n-place truth-function is a function from sequences of truth values to truth values, it can be represented by an unlabelled truth-table of 2^n rows (one row for each sequence of truth values of length n). It is unlabelled because the columns on the left are not headed by a sentence letter, so there is no association in the table between these strings of truth values and $\mathscr{L}_1$-structures. One such truth-function table is depicted in Table 4.2, for the truth-function **c**. This truth-function can also be represented perhaps more compactly as follows:

$$\mathbf{c}(x, y) = \begin{cases} F & \text{if } x = T \text{ and } y = F; \\ T & \text{otherwise.} \end{cases}$$

Theorem 47
There are 2^{2^n} n-place truth functions.

Proof. There are 2^n distinct sequences of truth values of length n, each of which is an input to an n-place truth function. Each of these sequences can be given T or F as its value. If one function assigns a different value to a given input than another function, they are different functions. So there are at least 2^{2^n} different truth functions (one for each way of assigning Fs and Ts to sequences of truth values of length n). And if f and g have the same value for every input, then f and g are the same function. (By extensionality of the underlying sets.) So there are exactly 2^{2^n} different n-place truth functions. □

Consider the 1-place truth functions. By Theorem 47, there are 4 such functions, outlined in Table 4.3. They are the *constant* functions **t** and **f**, which yield the same value for every argument; the *identity* function **i**, which yields as value the argument; and the *negation* function **n**, which yields the opposite truth value from the argument.

Expression We saw that we can represent a truth-function by an unlabelled truth-table. The truth-function table for **c** in Table 4.2 bears some resemblance to the truth table for $(P \to Q)$, but how can we characterise the relationship between truth-functions and labelled truth-tables for sentences more precisely? What we want to say is that a sentence ϕ is related to a truth-function f if, once we label the columns on the truth-function table in the standard way, we get the truth-table for ϕ. Here is the official definition.

Definition 73 (Expression). A sentence ϕ *expresses* a truth function f_n iff if ϕ contains exactly the sentence letters $s_1, \ldots, s_n$ (ordered in the standard way) then for every $\mathscr{L}_1$-structure $\mathscr{A}$,

$$f(\mathscr{A}(s_1), \ldots, \mathscr{A}(s_n)) = [\![\phi]\!]_{\mathscr{A}}\,.$$

So, for example, $(P \to Q)$ expresses the truth-function **c**, because it contains the sentence letters P, Q; the standard order on sentence letters places P before Q, and in every structure, $\mathbf{c}(\mathscr{A}(P), \mathscr{A}(Q)) = [\![(P \to Q)]\!]_{\mathscr{A}}$. Likewise, $(Q \to R)$ also expresses **c**. But $(Q \to P)$ does not express **c**, because in any structure $\mathscr{B}$ where $\mathscr{B}(P) = F, \mathscr{B}(Q) = T$, $[\![(Q \to P)]\!]_{\mathscr{B}} = F$ but $\mathbf{c}(\mathscr{B}(P), \mathscr{B}(Q)) = \mathbf{c}(F, T) = T$.

Since every $\mathscr{L}_1$ sentence has a truth-table produced in the orthodox way, every $\mathscr{L}_1$ sentence expresses some truth-function. The converse claim, that every truth-function is expressed by some sentence, will be proved in chapter 5 when we talk about *expressive adequacy* (page 81).

	t	**f**	**i**	**n**
T	T	F	T	F
F	T	F	F	T

Table 4.3: The four 1-place truth functions

Truth-Functional Connectives We've already seen some sentences which express interesting truth functions above: those sentences expressing the truth functions associated with the connectives of $\mathscr{L}_1$ (Table 4.1).

Definition 74 (Truth-functional Connective). A *truth-functional connective* is an n-place connective $\oplus$ such that there exists a n-place truth function f such that, for any $\phi_1, \ldots, \phi_n$,

$$[\![\oplus(\phi_1, \ldots, \phi_n)]\!]_{\mathscr{A}} = f\left([\![\phi_1]\!]_{\mathscr{A}}, \ldots, [\![\phi_n]\!]_{\mathscr{A}}\right),$$

i.e., if $\oplus(\phi_1, \ldots, \phi_n)$ expresses some truth function f.

It is evident from Table 4.1 that each of the connectives of $\mathscr{L}_1$ are truth-functional connectives, so $\mathscr{L}_1$ is a *truth-functional language*. We've already encountered the 1-place truth function corresponding to the negation connective: it is $\mathbf{n}$ from Table 4.3. We also encountered the 2-place connective $\mathbf{c}$ corresponding to the conditional connective. Here are three other pertinent two-place truth functions:

$$\mathbf{a}(x, y) = \begin{cases} F & \text{if } x = y = F; \\ T & \text{otherwise.} \end{cases}$$

$$\mathbf{k}(x, y) = \begin{cases} T & \text{if } x = y = T; \\ F & \text{otherwise.} \end{cases}$$

$$\mathbf{e}(x, y) = \begin{cases} T & \text{if } x = y; \\ F & \text{otherwise.} \end{cases}$$

It is easy to verify which connectives in Table 4.1 express these truth-functions.

Non-Truth Functional Connectives Consider the natural language connective 'necessarily', i.e., the connective that, given a sentence ϕ, yields a sentence ⌜Necessarily ϕ⌝. If ϕ is false in a structure, that obviously guarantees that ⌜Necessarily ϕ⌝ is false in that structure too (assuming the standard meaning for 'Necessarily'). But if ϕ is true in a structure, that doesn't tell us whether ⌜Necessarily ϕ⌝ is true in that structure: for some truths are necessary, others merely contingent. So to deal with the 'Necessarily' operator, we will have to introduce some further technology than

we have available in $\mathscr{L}_1$ structures and classical valuations. That is the topic taken up by *modal logic* – see the discussion at the end of chapter 13. We already can see, however, one expressive limitation of $\mathscr{L}_1$ as compared to English: for English includes non-truth-functional connectives like 'necessarily', 'probably', 'typically', etc., and $\mathscr{L}_1$ does not. Every sentence in $\mathscr{L}_1$ has its truth value determined by the valuation rules, and the valuation rules only make use of truth functions.

Revisiting the Semantics We could use our truth-functions to express the semantics for our language in another form, giving this alternative definition of a valuation.

Definition 75 (Alternative $\mathscr{L}_1$-valuation). When $\mathscr{A}$ is any $\mathscr{L}_1$-structure, the valuation function generated by $\mathscr{A}$ is this function:

$$[\![\phi]\!]_{\mathscr{A}} = \begin{cases} \mathscr{A}(\phi) & \text{if } \phi \text{ is a sentence letter;} \\ \mathbf{n}\left([\![\psi]\!]_{\mathscr{A}}\right) & \text{if } \phi = \neg\psi; \\ \mathbf{k}\left([\![\psi]\!]_{\mathscr{A}}, [\![\chi]\!]_{\mathscr{A}}\right) & \text{if } \phi = (\psi \wedge \chi); \\ \mathbf{a}\left([\![\psi]\!]_{\mathscr{A}}, [\![\chi]\!]_{\mathscr{A}}\right) & \text{if } \phi = (\psi \vee \chi); \\ \mathbf{c}\left([\![\psi]\!]_{\mathscr{A}}, [\![\chi]\!]_{\mathscr{A}}\right) & \text{if } \phi = (\psi \rightarrow \chi); \\ \mathbf{e}\left([\![\psi]\!]_{\mathscr{A}}, [\![\chi]\!]_{\mathscr{A}}\right) & \text{if } \phi = (\psi \leftrightarrow \chi). \end{cases}$$

This definition is obviously equivalent to that in Definition 59, in that both definitions determine the same valuation of all the sentences of the language given the same initial structure.

Definition 76 (Base). (Beall and van Fraassen, 2003: 27) Let $[\![\cdot]\!]_{\mathscr{A}}$ be a classical valuation. A *base* $\mathfrak{B}$ for $[\![\cdot]\!]_{\mathscr{A}}$ is a triple $\langle V, \mathbb{O}, c\rangle$ where V is a set of elements, $\mathbb{O}$ a set of operators on V (functions such that their domain and range are both V – recall Definition 39), and c is a function from the set of connectives of $\mathscr{L}_1$, $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, into $\mathbb{O}$, such that

- $[\![\phi]\!]_{\mathscr{A}} \in V$, for any $\mathscr{L}_1$ sentence ϕ;
- for any sentences $\phi_1, \ldots, \phi_n$, and any n-ary connective $\oplus$,

$$[\![\oplus(\phi_1, \ldots, \phi_n)]\!]_{\mathscr{A},\mathscr{L}} = c(\oplus)\left([\![\phi_1]\!]_{\mathscr{A},\mathscr{L}}, \ldots, [\![\phi_n]\!]_{\mathscr{A},\mathscr{L}}\right).$$

Informally, a base for a valuation is a set of values V, a set of truth-functions $\mathbb{O}$, and a mapping that takes connectives of the language to the truth-functions they express (Definition 74). A base for $\mathscr{L}_1$ is this: let $V = \{T, F\}$, $\mathbb{O} = \{\mathbf{n}, \mathbf{k}, \mathbf{a}, \mathbf{c}, \mathbf{e}\}$, and let c be the function that maps $\neg$ to $\mathbf{n}$, $\wedge$ to $\mathbf{k}$, *etc.*

Exercises

1. Show the *reductio* rule: that $\Gamma \vDash$ iff for every $\gamma \in \Gamma$, $\Gamma \setminus \{\gamma\} \vDash \neg\gamma$. (Hint: Theorem 26.)
2. State what features the truth table for $\Gamma \cup \{\phi, \psi\}$ will possess when
   (a) Γ entails ϕ;
   (b) Γ is consistent;
   (c) ϕ is a contradiction;
   (d) ϕ and ψ are logically equivalent.
3. (a) Show that $\vDash \phi \leftrightarrow \psi$ iff $Cn(\{\phi\}) = Cn(\{\psi\})$.
   (b) Prove Lemma 27.
   (c) Prove Lemma 29.
4. Show the remaining clauses of the induction step for the proof of Theorem 23 for the cases where the complex sentence is of the forms:
   (a) $(\phi \wedge \psi)$;
   (b) $(\phi \rightarrow \psi)$;
   (c) $(\phi \leftrightarrow \psi)$.
5. Prove Equivalents (Theorem 45).
6. (a) Prove the remaining induction cases in the proof of Equivalence (Theorem 46).
   (b) As a corollary of the Equivalence Theorem, prove that if $\vDash \phi \leftrightarrow \psi$ and ϕ is a constituent sentence of χ, then $\vDash \chi \leftrightarrow \chi[\psi/\phi]$.
7. Show that, if $\phi, \psi \vDash \chi$ is a false sequent, it has a substitution instance $\phi', \psi' \vDash \chi'$, in which ϕ' and ψ' are tautologies and χ' is inconsistent. (Note: by 'substitution instance' in the question, you are to understand a sequent that results from one *or more* uniform substitutions of sentence letters. It follows from General Substitution [Theorem 44] that successive chains of uniform substitutions never allow one to transform a correct sequent to an incorrect one.)
8. (a) Prove Transitivity.

(b) Prove Contraposition.

(c) Prove this rule, related to Cut: if $\Gamma \vDash \phi$, and $\phi, \Delta \vDash \psi$, then $\Gamma, \Delta \vDash \psi$.

9. Suppose ϕ and ψ contain the same sentence letters. Prove that ϕ and ψ are logically equivalent iff they express the same truth function.

10. Consider the deprived language which has the same syntactic formation rules and semantics as $\mathscr{L}_1$, but only contains the connectives $\neg$ and $\leftrightarrow$. Show that, in this language, every sentence is logically equivalent to a sentence in which no occurrence of $\neg$ has $\leftrightarrow$ in its scope.

11. (a) Show the *basic principle for disjunction* (Bostock, 1997: §2.5):

$$\Gamma, \phi \vee \psi \vDash \text{ iff } \Gamma, \phi \vDash \text{ and } \Gamma, \psi \vDash .$$

(b) Show that the basic principle for disjunction implies that $\phi \vDash \phi \vee \psi$ and $\psi \vDash \phi \vee \psi$.

(c) Show that the basic principle for disjunction implies that if (i) $\Gamma, \phi \vDash \chi$ and (ii) $\Delta, \psi \vDash \chi$ then (iii) $\Gamma, \Delta, \phi \vee \psi \vDash \chi$.

Answers to selected exercises on page 241.

Chapter 5

Metatheory of $\mathscr{L}_1$

5.1 Disjunctive Normal Form

Definition 77 (Arbitrary Conjunction). An inductive definition:

- $\left[\bigwedge_{i=1}^{1} \phi_i\right] =_{\text{df}} \phi_1$;
- $\left[\bigwedge_{i=1}^{n+1} \phi_i\right] =_{\text{df}} \left(\left[\bigwedge_{i=1}^{n} \phi_i\right] \wedge \phi_{n+1}\right)$.

The definition of arbitrary disjunction, $\bigvee_{i=1}^{n}$, is wholly parallel to that for arbitrary conjunction. (Giving it explicitly is left as an exercise.)

Definition 78 (Literal). A *literal* is any sentence letter or negated sentence letter.

Definition 79 (Disjunctive Normal Form (DNF)). An $\mathscr{L}_1$ sentence of ϕ is in *disjunctive normal form* iff there exist $n, m_1, \ldots, m_n$ such that

$$\phi = \left[\bigvee_{i=1}^{n} \left[\bigwedge_{j=1}^{m_i} \pm s_{i,j}\right]\right],$$

where each $\pm s_{i,j}$ is a literal.

That is: a sentence is in disjunctive normal form iff it is a (perhaps degenerate) disjunction of (perhaps degenerate) conjunctions of (perhaps negated) sentence letters. One clear example is $((P \wedge \neg Q) \vee (P \wedge Q))$. But since P is a literal, and

it is a degenerate arbitrary conjunction, and a degenerate arbitrary disjunction, it is in DNFalso. Another way of characterising which $\mathscr{L}_1$ sentences are in DNFis as follows: a sentence is DNFiff it contains at most only connectives drawn from $\{\neg, \wedge, \vee\}$, and such that $\vee$ never occurs in the scope of $\wedge$ or $\neg$, and $\wedge$ never occurs in the scope of $\neg$.

Our interest in disjunctive normal form lies in the following theorem:

Theorem 48 (DNF)
Every truth function is expressed by a $\mathscr{L}_1$ sentence in DNF.

Proof. Suppose f is an n-place truth function. Let $s_1, \ldots, s_n$ be sentence letters in the standard order, and let $\mathscr{A}$ be any structure. Define an *f-agreeing* structure $\mathscr{A}$ as one where $f(\llbracket s_1 \rrbracket_{\mathscr{A}}, \ldots, \llbracket s_n \rrbracket_{\mathscr{A}}) = T$. Assume there is at least one f-agreeing structure. Then define

$$S_i^{\mathscr{A}} = \begin{cases} s_i & \text{if } \llbracket s_i \rrbracket_{\mathscr{A}} = T; \\ \neg s_i & \text{if } \llbracket s_i \rrbracket_{\mathscr{A}} = F. \end{cases}$$

It is obvious that $\left\llbracket S_i^{\mathscr{A}} \right\rrbracket_{\mathscr{A}} = T$ for each $\mathscr{A}, i$.

For every structure $\mathscr{A}$, define $\mathfrak{c}_{\mathscr{A}} = S_1^{\mathscr{A}} \wedge \ldots \wedge S_n^{\mathscr{A}} = \left[\bigwedge_{i=1}^{n} S_i^{\mathscr{A}}\right]$. Again, it is obvious that $\llbracket \mathfrak{c}_{\mathscr{A}} \rrbracket_{\mathscr{A}} = T$, since it is a conjunction of true conjuncts. It is equally obvious that for any structure $\mathscr{B}$ that doesn't agree with $\mathscr{A}$ on the sentence letters $s_1, \ldots, s_n$, $\llbracket \mathfrak{c}_{\mathscr{A}} \rrbracket_{\mathscr{B}} = F$, since it will have at least one false conjunct.

Since there are only finitely many structures that differ from one another in their assignments to $s_1, \ldots, s_n$, there are only finitely many distinct sentences $\mathfrak{c}_{\mathscr{A}}$ for various $\mathscr{A}$. Therefore, there are only finitely many $\mathfrak{c}_{\mathscr{A}}$ *such that $\mathscr{A}$ is an f-agreeing structure*; let them be enumerated $\mathfrak{c}_1, \ldots, \mathfrak{c}_m$. It is obvious that each $\mathfrak{c}_i$ is true in one and only one structure, and that each such structure is f-agreeing.

Define $\mathfrak{d} = \mathfrak{c}_1 \vee \ldots \vee \mathfrak{c}_m = \left[\bigvee_{j=1}^{m} \mathfrak{c}_j\right]$, *unless* there were no f-agreeing structures, in which case let $\mathfrak{d} = (P \wedge \neg P)$. It is obvious that $\mathfrak{d}$ is true in any structure which is among those such that some $\mathfrak{c}_i$ is true, i.e., $\mathfrak{d}$ is true in any f-agreeing structure.

$\mathfrak{d}$ is in DNF, by construction; and $\llbracket \mathfrak{d} \rrbracket_{\mathscr{A}} = f(\llbracket s_1 \rrbracket_{\mathscr{A}}, \ldots, \llbracket s_n \rrbracket_{\mathscr{A}})$ for all $\mathscr{A}$, because it captures all and only the f-agreeing structures. So $\mathfrak{d}$ expresses f. □

The truth-table rationale for this proof is clear: first, find the rows of the truth table on which the truth-function gets T (those are the f-agreeing rows). Specify a sentence true in exactly one row by conjoining the relevant literals (so if the row

corresponding to $\mathscr{A}$ is $[\![P]\!]_{\mathscr{A}} = T$, $[\![Q]\!]_{\mathscr{A}} = F$, the relevant conjunction is $P \wedge \neg Q$). Then disjoin those conjunctions which correspond to the f-agreeing rows; the result is a DNF sentence true in exactly the f-agreeing rows.

We can use this result to show this

Corollary 49

For any sentence ϕ, there is a sentence ϕ' which is logically equivalent to ϕ and which is in DNF form.

Proof. This follows immediately from the DNF theorem and the fact that every sentence expresses a truth function. □

CNF and Positive Truth Functions A sentence is said to be in *Conjunctive Normal Form* iff it is a (possibly degenerate) conjunction of (possibly degenerate) disjunctions of literals. This is obviously a dual notion to DNF. In the case of DNF, the DNF sentence expressing some truth function f is the disjunction of conjunctions, each of which specifies an f-agreeing structure. Each conjunct is therefore *sufficient* for f to have the value T in the corresponding structure; the DNF sentence is the disjunction of all of these sufficient conditions. For CNF, we want the dual notion: the CNF expression of f will be a sentence which is a conjunction of disjunctions, each of which specifies a necessary condition for a structure to be f-agreeing. And what is a necessary condition for a structure to be f-agreeing? It must avoid being *f-disagreeing* – so each disjunct will be a disjunction of sentence letters and negated sentence letters, the truth of any of which is sufficient to ensure that some f-disagreeing structure is avoided. (A problem asks you to make the foregoing remarks into a more precise proof of the CNF theorem.)

Call a truth function f *positive* iff $f(T, \ldots, T) = T$ (Bostock, 1997: exercise 2.9.3). (Equivalently, if the top row of the truth-function table is T.) We can show that all truth-functions which can be expressed using only $\rightarrow$ and $\wedge$ are positive (left for exercises). Of interest is the converse theorem:

Theorem 50

All positive truth functions can be expressed by $\rightarrow$ and $\wedge$.

Proof. I sketch the proof. Suppose the contrary, for *reductio*. Then there is a truth function f which is positive but can't be expressed by $\wedge, \rightarrow$. There is a CNF sen-

tence ϕ_c which expresses f; this will be a conjunction of disjunctions of (possibly negated) sentence letters. Note the following results:

- $\phi \vee \psi$ is logically equivalent to $(\psi \to \phi) \to \phi$,
- $\phi \vee \neg\psi$ is logically equivalent to $\psi \to \phi$, and
- $\neg\phi \vee \psi$ is logically equivalent to $\phi \to \psi$.

By the constructive proof of the CNF theorem, it can be shown – by induction on complexity of sentences, and using these results – that each conjunct of the resulting CNF sentence will be equivalent to either

1. an arrow sentence *without* negation, or
2. a conjunct of the form $\bigvee_i \neg s_i$ for the sentence letters in ϕ_c (exercise).

In the second case, the CNF sentence as a whole can't be positive (because a disjunction of negated sentence letters will be false on the top row of the truth table, so the conjunction will be false). So the first case must obtain for ϕ_c – each of its conjuncts must be logically equivalent to a negation-free (possibly nested) conditional. By substituting each conjunct of ϕ_c for the arrow-sentence equivalent, we obtain a sentence ϕ, which is a conjunction of arrow sentences without negation. But ϕ expresses f and contains only $\to$ and $\wedge$. □

5.2 Expressive Adequacy and Functional Completeness

Expressive Adequacy and Functional Completeness

Definition 80 (Expressive Adequacy). A set of connectives is *expressively adequate* iff there is, for any truth function f, a sentence containing only those connectives which expresses f.

The DNF theorem shows that $\{\vee, \wedge, \neg\}$ is expressively adequate, and therefore that the set of connectives of $\mathscr{L}_1$ is expressively adequate. A language is by extension also said to be expressively adequate iff the connectives it contains are expressively adequate there is, for each truth function f, some sentence in that language

that, under the intended semantics, expresses f. Since $\mathscr{L}_1$ sentences in DNF form are collectively expressively adequate, $\mathscr{L}_1$ is expressively adequate.

Definition 81 (Functional Completeness). A set of truth-functional operators (truth-functions) $\mathbb{O}$ is *functionally complete* if every truth function can be constructed by (possibly nested) combination of operators from only those in $\mathbb{O}$.[1]

Derivatively, a set of *connectives* is functionally complete if the functions expressed by those connectives are jointly functionally complete.

The DNF theorem shows that every truth function is expressed by an $\mathscr{L}_1$ sentence, and so $\mathscr{L}_1$ is expressively adequate. It may easily be adapted to show that the set of operators $\{\mathbf{n}, \mathbf{k}, \mathbf{a}\}$ is functionally complete in the strict sense. First we introduce some generalisations of $\mathbf{k}$ and $\mathbf{a}$, which will allow us to apply functions to arbitrary sequences of truth values:

Definition 82 (Maximum and Minimum). The *maximum* of a sequence of truth values is defined inductively:

- $\max_{i=1}^{1} v_i = v_1$;
- $\max_{i=1}^{n+1} v_i = \mathbf{a}(\max_{i=1}^{n}, v_{n+1})$.

The *minimum* of a sequence of truth values is defined:

- $\min_{i=1}^{1} v_i = v_1$;
- $\min_{i=1}^{n+1} v_i = \mathbf{k}(\min_{i=1}^{n}, v_{n+1})$.

According to this definition, $\max(T, F) = T$ and $\min(T, F) = F$.

Theorem 51 (Functional Completeness of $\{\mathbf{n}, \mathbf{k}, \mathbf{a}\}$)
If $f(x_1, \ldots, x_n)$ is an n-place truth function, there is some function composed from max, min, *and* $\mathbf{n}$ *which is identical to f (has the same values for the same arguments), and is constructed in such a way that (i)* max *takes scope over any occurrence of* min *and* $\mathbf{n}$ *in the function definition, and (ii)* min *takes scope over any occurence of* $\mathbf{n}$ *in the function definition.*

[1] Remember that each truth-function is an operator since it is a function from sequences of truth values to truth values.

Proof. This example illustrates the main idea, which is just that involved in the proof of the DNF theorem.

Suppose we consider the binary function f such that $f(x, y) = F$ iff $x = F, y = T$. Then $f(x, y) = T$ iff either $x = y = T$ or $x = \mathbf{n}(y) = T$ or $\mathbf{n}(x) = \mathbf{n}(y) = T$. So

$$f(x, y) = \max(\min(x, y), \min(x, \mathbf{n}(y)), \min(\mathbf{n}(x), \mathbf{n}(y))).$$ □

That is, any truth-function expressible by a sentence in DNF can be constructed from the truth-functions associated with (generalised) disjunction, (generalised) conjunction, and negation; since every truth-function can be expressed by a sentence in DNF, those operators are collectively functionally complete. Accordingly, the connectives which express those operators $\{\neg, \wedge, \vee\}$ form a functionally complete set (in the derivative sense).

What this last result shows is that, in some sense, $\rightarrow$ and $\leftrightarrow$ are *dispensible* – every truth function expressible using them can be expressed by a sentence without them (i.e., a DNF sentence or a CNF sentence).

There is another way to show this same result. Since two sentences containing the same stock of sentence letters are logically equivalent iff they express the same truth-function, the DNF theorem, together with the fact that every $\mathscr{L}_1$ sentence expresses a truth-function, suffices to show that every $\mathscr{L}_1$ sentence is logically equivalent to one in DNF form. Hence sentences involving $\rightarrow$ and $\leftrightarrow$ are logically equivalent to sentences in DNF form, and therefore in a sense dispensible – we can't say anything more with conditional and biconditional sentences than we could if we had restricted ourselves to sentences in DNF.[2]

The De Morgan Equivalences Let '$\phi \dashv\vDash \psi$' abbreviate '$\phi \vDash \psi$ and $\psi \vDash \phi$' – that is, that ϕ and ψ are logically equivalent.

Theorem 52 (De Morgan Equivalences)

$$(\phi \vee \psi) \dashv\vDash \neg(\neg\phi \wedge \neg\psi),$$
$$(\phi \wedge \psi) \dashv\vDash \neg(\neg\phi \vee \neg\psi).$$

[2]It is more efficient to say $P \leftrightarrow Q$ than its DNF equivalent $(P \wedge Q) \vee (\neg P \wedge \neg Q)$.

ϕ	ψ	$\phi \wedge \psi$	$\neg(\neg\phi \vee \neg\psi)$	$\phi \vee \psi$	$\neg(\neg\phi \wedge \neg\psi)$
T	T	T	T	T	T
T	F	F	F	T	T
F	T	F	F	T	T
F	F	F	F	F	F

Table 5.1: De Morgan Equivalences

Proof. These equivalences can be easily demonstrated using a schematic truth table (Table 5.1). □

These equivalences show that $\{\neg, \vee\}$ and $\{\neg, \wedge\}$ are expressively adequate, given the DNF theorem.

There are analogues of these equivalences for truth-functions:

Theorem 53 (De Morgan Equivalences for Truth Functions)

$$\mathbf{a}(x, y) = \mathbf{n}(\mathbf{k}(\mathbf{n}(x), \mathbf{n}(y))),$$
$$\mathbf{k}(x, y) = \mathbf{n}(\mathbf{a}(\mathbf{n}(x), \mathbf{n}(y))).$$

Using this result we can establish Theorem 52 without use of a truth table.[3] This result also shows that $\{\mathbf{n}, \mathbf{k}\}$ and $\{\mathbf{n}, \mathbf{a}\}$ are functionally complete.

Inadequacy and New Connectives A set of connectives is *expressively inadequate* iff there is some truth function which cannot be expressed by sentences involving only those connective. Accordingly, $\{\neg\}$ is expressively inadequate: quite

[3]Consider the equivalence of $(\phi \vee \psi)$ and $\neg(\neg\phi \wedge \neg\psi)$. By Theorem 53, $\mathbf{a}(x, y) = \mathbf{n}\left(\mathbf{k}\left(\mathbf{n}\left(x\right), \mathbf{n}\left(y\right)\right)\right)$, so for any $\mathscr{A}$,

$$\begin{aligned}
[\![\phi \vee \psi]\!]_{\mathscr{A}} &= \mathbf{a}\left([\![\phi]\!]_{\mathscr{A}}, [\![\psi]\!]_{\mathscr{A}}\right) \\
&= \mathbf{n}\left(\mathbf{k}\left(\mathbf{n}\left([\![\phi]\!]_{\mathscr{A}}\right), \mathbf{n}\left([\![\psi]\!]_{\mathscr{A}}\right)\right)\right) \\
&= \mathbf{n}\left(\mathbf{k}\left([\![\neg\phi]\!]_{\mathscr{A}}, [\![\neg\psi]\!]_{\mathscr{A}}\right)\right) \\
&= \mathbf{n}\left([\![(\neg\phi \wedge \neg\psi)]\!]_{\mathscr{A}}\right) \\
&= [\![\neg(\neg\phi \wedge \neg\psi)]\!]_{\mathscr{A}}.
\end{aligned}$$

ϕ	ψ	$(\phi \uparrow \psi)$	$(\phi \downarrow \psi)$
T	T	F	F
T	F	T	F
F	T	T	F
F	F	T	T

Table 5.2: Sheffer Stroke and Peirce Arrow

apart from whether it can express a 2- or more place truth function, consider the 1-place function $\mathbf{t}$ which is constantly true. Since $\mathbf{t}(T) = \mathbf{t}(F)$, we would need a sentence ϕ involving only negation such that in even one case $[\![\phi]\!]_{\mathscr{A}} = [\![\neg\phi]\!]_{\mathscr{A}}$; this clearly cannot be.

A set of truth functions is *functionally incomplete* iff there is some truth function which is not constructible from functions in that set. The examples just considered show that $\{\mathbf{n}\}$ is functionally incomplete; since $\neg$ cannot express $\mathbf{t}$, so $\mathbf{t}$ can't be constructed solely from $\mathbf{n}$.

Given our definition of a connective, we can imagine extending our language $\mathscr{L}_1$ by adding new sentence-forming operators that express different truth functions. (We need to extend the formation rules in the syntax accordingly too, but that is straightforward once we know the arity of the new connectives.) We could add the 0-place connective $\top$ – this is like a sentence letter with a fixed interpretation, T in every structure. As we know from the DNF theorem, adding $\top$ doesn't add to the expressive power of the language: $\top$ is logically equivalent to $(P \vee \neg P)$, and so adding it would be redundant. Adding $\wedge$ to the language $\mathscr{L}_\neg$, in which the only connective is negation, is not redundant: $\mathscr{L}_{\neg,\wedge}$ is functionally complete and $\mathscr{L}_\neg$ is not.

Sheffer Stroke and Peirce Arrow The connectives $\uparrow$ and $\downarrow$ can be added to a language, with the truth tables in Table 5.2. The $\uparrow$, sometimes also written $|$, is called the *Sheffer stroke*; the $\downarrow$ has no standard name, but is sometimes called the *Peirce arrow*.

Each of these connectives is expressively adequate. The Sheffer stroke is: $\neg\phi$ is logically equivalent to $(\phi \uparrow \phi)$, and $(\phi \wedge \psi)$ is logically equivalent to $((\phi \uparrow \psi) \uparrow (\phi \uparrow \psi))$. The case of the Peirce arrow is left for an exercise.

Definition 83 (nand and nor).

$$\mathbf{nand}(x, y) =_{\text{df}} \begin{cases} F \text{ if } x = y = T; \\ T \text{ otherwise.} \end{cases}$$

$$\mathbf{nor}(x, y) =_{\text{df}} \begin{cases} T \text{ if } x = y = F; \\ F \text{ otherwise.} \end{cases}$$

Each of these truth functions is by itself functionally complete. E.g., we may construct **n** and **k** using just **nand**:

$$\mathbf{n}(x) = \mathbf{nand}(x, x); \text{ and}$$
$$\mathbf{k}(x, y) = \mathbf{nand}(\mathbf{nand}(x, y), \mathbf{nand}(x, y)).$$

Obviously $\uparrow$ expresses **nand** and $\downarrow$ expresses **nor**.

5.3 Duality

Duality $\wedge$ and $\vee$, as well as $\uparrow$ and $\downarrow$, are what is known as *duals* of one another.

Definition 84 (Duality). The *dual* of a truth-functional connective is that connective whose truth table results from that of the given connective by replacing *every* occurrence of T by F and every occurrence of F by T (see Table 5.3[4]).

Other connectives have duals too: $\phi \leftrightarrow \psi$ is dual to the operator $\nleftrightarrow$ (where $\phi \nleftrightarrow \psi$ is true just in case ϕ and ψ have different truth values). Negation $\neg$ is an interesting

ϕ	ψ	$\phi \wedge \psi$	$\phi \uparrow \psi$
T	T	T	F
T	F	F	F
F	T	F	F
F	F	F	T

$\Leftrightarrow$

ϕ	ψ	$\phi \vee \psi$	$\phi \downarrow \psi$
F	F	F	T
F	T	T	T
T	F	T	T
T	T	T	F

Table 5.3: Duality Illustrated Using Truth Tables

[4]Note that the right hand truth table is written upside down, but obviously the order of the rows doesn't matter to a truth table.

case: it is its own dual, as can readily be verified by transforming the truth table for $\neg P$ appropriately.

The Duality Theorem for $\wedge, \vee$ Suppose that ϕ is a sentence of $\mathscr{L}_1$ which involves only connectives in $\{\wedge, \vee, \neg\}$ (any sentence of $\mathscr{L}_1$ will be equivalent to some such ϕ). Now we define two operations on sentences of this form:

1. Let $\overline{\phi}$ be the sentence that results from uniformly substituing $\ulcorner\neg s\urcorner$ for each sentence letter $\ulcorner s\urcorner$ in ϕ (even those already negated).

2. Let $\phi^\star$ be the sentence that results from replacing each connective in ϕ by its dual.

Note to begin that $\neg neg\overline{\phi} = \overline{\neg\phi}$ and $\overline{\phi} \wedge \overline{\psi} = \overline{\phi \wedge \psi}$, because the overlining operation applies only to sentence letters – it doesn't matter whether we apply it to the constituents first then apply the connectives, or apply the connectives first then apply the overlining operation.

Lemma 54 (Duality for Conjunction and Disjunction)
For any ϕ, $\phi^\star \dashv\vDash \neg\overline{\phi}$.

Proof. Base case: ϕ is just a sentence letter s: so that $\phi^\star = s$, and $\neg\overline{\phi} = \neg\neg s$, which are equivalent.

Induction step:

1. ϕ is $\neg\psi$, and the hypothesis holds for ψ. Then $\phi^\star = (\neg\psi)^\star = \neg(\psi^\star)$, because $\neg$ is its own dual. Because the theorem holds of ψ, $\neg(\psi^\star) \dashv\vDash \neg\neg\overline{\psi}$; but $\neg\neg\overline{\psi} = \neg\overline{\neg\psi} = \neg\overline{\phi}$.

2. ϕ is $(\psi \wedge \chi)$, and the hypothesis holds for ψ, χ. $\phi^\star = (\psi \wedge \chi)^\star = \psi^\star \vee \chi^\star$. By the induction hypothesis, $\psi^\star \vee \chi^\star \dashv\vDash \neg\overline{\psi} \vee \neg\overline{\chi}$. By De Morgan, $\neg\overline{\psi} \vee \neg\overline{\chi} \dashv\vDash \neg(\overline{\psi} \wedge \overline{\chi})$, which is $\neg(\overline{\psi \wedge \chi}) = \neg(\overline{\phi})$.

3. ϕ is $(\psi \vee \chi)$; much as for $\wedge$. □

Theorem 55 (Duality for Conjunction and Disjunction)
If $\phi^\star$ is dual to ϕ and $\psi^\star$ is dual to ψ, then $\phi \vDash \psi$ iff $\psi^\star \vDash \phi^\star$.

Proof. Suppose $\phi \vDash \psi$. Then, by Substitution of each sentence letter by its negation, $\overline{\phi} \vDash \overline{\psi}$. By Contraposition, $\neg\overline{\psi} \vDash \neg\overline{\phi}$. By the Duality Lemma, $\psi^\star \vDash \phi^\star$. □

Corollary 56
If $\phi \dashv\vDash \psi$, $\phi^\star \dashv\vDash \psi^\star$.

Each De Morgan equivalence in Theorem 52 follows from the other, given Corollary 56.

Theorem 57 (Duality for Tautologies)
If $\vDash \phi$, *then* $\vDash \neg(\phi^\star)$.

Proof. Suppose $\vDash \phi$. Since ϕ is true in every structure, $\overline{\phi}$ is also true in every structure (obvious by substitution). So $\vDash \overline{\phi}$. By the Duality Lemma, $\overline{\phi} \dashv\vDash \neg\phi^\star$. So $\vDash \neg\phi^\star$. □

Duality Generalised Every truth functional connective $\oplus$ – in any truth-functional language – has a dual, which we denote $\oplus^\star$ by extending our previous notation. A set of truth functors $\mathbb{C} = \{\oplus_1, \ldots, \oplus_n\}$ is *self-dual* iff $\mathbb{C} = \{\oplus_1^\star, \ldots, \oplus_n^\star\}$. It is obvious that $\{\neg\}$ is self-dual. Given the result above, the set $\{\wedge, \vee\}$ is self-dual, as $\vee = \wedge^\star$ and $\wedge = \vee^\star$.

If $\mathscr{A}$ is a structure, let $\mathscr{A}^\star$ be the structure such that for all ϕ, $\mathscr{A}^\star(\phi) = T^\star$ iff $\mathscr{A}(\phi) = T$. Consider just those languages $\mathscr{L}$ such that a connective $\oplus$ is in $\mathscr{L}$ iff $\oplus^\star$ is in $\mathscr{L}$. ($\mathscr{L}_1$is among these languages.) The dual $\phi^\star$ of a sentence ϕ of $\mathscr{L}$ results from replacing *every* connective in ϕ by its dual. Then we can show:

$$[\![\phi]\!]_{\mathscr{A}} = \mathbf{n}\left([\![\phi^\star]\!]_{\mathscr{A}^\star}\right).$$

5.4 Interpolation

Craig Interpolation Theorem for Sentential Logic

Theorem 58 (Craig Interpolation)
If $\phi \vDash \psi$, *and there is a non-empty set* B *of sentence letters occurring in both* ϕ *and* ψ, *then there is a sentence* χ *(an* interpolant*) such that both* $\phi \vDash \chi$ *and* $\chi \vDash \psi$, *and each sentence letter in* χ *is in* B.

Proof. Let the members of $\boldsymbol{B}$ be enumerated $b_1, \ldots, b_n$.

Let $\{\mathscr{B}_1, \ldots, \mathscr{B}_{2^n}\}$ be a set of structures every pair of which differ on their assignment of a truth value to at least one $b \in \boldsymbol{B}$. Define a n-place truth function f_χ, for every structure $\mathscr{B}_i$:

$$f_\chi\left(\left\langle |b_1|_{\mathscr{B}_i}, \ldots, |b_n|_{\mathscr{B}_i}\right\rangle\right) = \begin{cases} & \text{if there is a structure } \mathscr{A} \text{ which} \\ F & \text{agrees with } \mathscr{B}_i \text{ on } \boldsymbol{B} \text{ in which} \\ & [\![\psi]\!]_{\mathscr{A}} = F; \\ T & \text{otherwise.} \end{cases}$$

By construction, f_χ is a total truth function. The sentence χ that (by DNF) expresses f_χ is our needed interpolant.

- χ clearly entails ψ, since by construction no structure that makes χ true makes ψ false (since every structure agrees with some $\mathscr{B}_i$ on $\boldsymbol{B}$).

- Moreover, χ is entailed by ϕ. Suppose ϕ is true in some structure $\mathscr{C}$, but χ is not. By definition, χ is false in $\mathscr{C}$ because (i) either ψ is false in $\mathscr{C}$, or (ii) ψ is false in a structure just like $\mathscr{C}$ in what it assigns to the members of $\boldsymbol{B}$. Option (i) is excluded, because $\phi \vDash \psi$. So if χ is false in $\mathscr{C}$, it must be because some structure agreeing with $\mathscr{C}$ on $\boldsymbol{B}$ makes ψ false. This structure need differ from $\mathscr{C}$ only in what it assigns to sentence letters in ψ that are not in ϕ, because obviously it is some difference in those sentence letters that makes ψ true in $\mathscr{C}$ and false in the other structure. But then the other structure need not differ from $\mathscr{C}$ at all in the truth values of sentence letters in ϕ; and we may therefore assume it does not. So that structure must make ϕ true – since $\phi \vDash \psi$, ψ is both true and false in that structure. Contradiction; so there is no such $\mathscr{C}$, by *reductio*. But $\mathscr{C}$ was arbitrary, so $\phi \vDash \chi$. □

The Craig Interpolation Theorem actually names the version of this result proved for predicate logic (Craig, 1957).

Corollary 59

If ϕ is logically equivalent to ψ, and there is a non-empty set $\boldsymbol{B}$ of sentence letters occuring in both ϕ and ψ, then there is a sentence χ logically equivalent to both ϕ and ψ such that each sentence letter in χ is in $\boldsymbol{B}$.

Proof. Straightforward from transitivity of entailment and Theorem 58. □

There was some philosophical rationale behind interest in interpolation theorems. The *instrumentalists* were a group of philosophers of science who were sceptical of unobservable theoretical entities, to the point where they thought there was no good way of understanding a language which purported to refer to such entities. So they said that terms like 'electron', which is a theoretical term referring to an entity whose existence is inferred from observation of intermediaries rather than being observed directly, are strictly speaking meaningless. One might worry whether this cripples our science: what sort of particle physics could we do if we had to get rid of the word 'electron'? Craig's Interpolation Theorem offers some help here. Suppose that **T** is sentence of (formalised) physics, including subsentences which purport to refer to electrons, etc. And suppose **O** is some purely observational consequence of **T** – the sort of claim that might be used to test the theory, or make a practical difference. Then the Craig Interpolation Theorem says there is a sentence **T/E** *purely using observational terms* (since those are the only sort that occur in **O**) that is a consequence of **T** and which entails **O**. **T/E** is, in some sense, the purely observational *empirical* content of **T**. And – said the instrumentalists – this is great! For every sentence of physics which makes reference to theoretical entities, we can come up with a theoretically hygenic sentence which does not, which is also a consequence of the theory, but which is framed in purely observational vocabulary.

But note that the interpolated sentence isn't a hygienic as it was supposed. For suppose we have a sentence in our observation language that means *everything is observable*. We have negation; so there will be a sentence true iff something is unobservable. This is supposedly expressed in the 'acceptable' vocabulary, but of course it makes a claim about purely theoretical unobservable entities. So the philosophical significance of the interpolated sentence seems negligible:

> The empirical import of a theory cannot be isolated in this syntactical fashion, by drawing a distinction among theorems in terms of vocabulary. If that could be done, **T/E** would say exactly what **T** says about what is observable and what it is like, and nothing more. But any unobservable entity will differ from the observable ones in the way it systematically lacks observable characteristics. As long as we do not abjure negation, therefore, we shall be able to state in the observa-

> tional vocabulary (however conceived) that there are unobservable entities, and, to some extent, what they are like. The quantum theory, Copenhagen version, implies that there are things which sometimes have a position in space, and sometimes have not. This consequence I have just stated without using a single theoretical term. Newton's theory implies that there is something (to wit, Absolute Space) which neither has a position nor occupies a volume. Such consequences are by no stretch of the imagination about what there is in the observable world, nor about what any observable thing is like. The reduced [interpolated] theory **T/E** is not a description of part of the world described by **T**; rather, **T/E** is, in a hobbled and hamstrung fashion, the description by **T** of everything. (van Fraassen, 1980: 54–5)

5.5 Compactness

Theorem 58 is interesting regardless of philosophical programs it may have been associated with. It tells us that, to understand what is going on in an argument, we need only consider the sentence letters in common to premises and conclusion. Since there are only finitely many of those, the correctness or otherwise of entailments between sentences reduces to a computationally tractable problem: look at the (finitely many) sentences that constitute the premise and conclusion; determine the sentence letters in common; and then see if you can construct an interpolant.

Infinitely Many Premises But this procedure fails at the first step if we allow arbitrary sets of premises on the left hand side of the entailment. If the set of premises is finite, then we can simply consider the conjunction of all its members, which will be finite. But what happens to the notion of an interpolant, and the idea that entailment is always mediated only by sentence letters occurring in both premises and conclusion, if the set of premises is infinite?

This is not merely a theoretical worry. Consider this argument in English:

(1) There is at least 1 thing.

(2) There are at least 2 things.

(3) There are at least 3 things.

⋮

(*n*) There are at least *n* things.

⋮

(C) There are infinitely many things.

This argument is valid. However, it needs to make essential use of infinitely many of its premises. For any *finite* set of premises of this argument can be jointly true while the conclusion is false.[5]

Compactness English is thus a language in which there are valid arguments with infinitely many premises, but where no finite subset of those premises entails the conclusion. Let us introduce a technical term.

Definition 85 (Compactness). An interpreted language $\mathscr{L}$ is said to be *compact* iff any unsatisfiable infinite set of sentences of $\mathscr{L}$ has a finite unsatisfiable subset. Contrapositively: if every finite subset of some Γ is satisfiable, then Γ is satisfiable.

Since Γ entails ϕ iff $\Gamma \cup \{\neg\phi\}$ is unsatisfiable, English is *not* a compact language. The union of all the premises stating that there are at least *n* things together with the claim that there are only finitely many things is unsatisfiable, yet any finite subset of that set must contain at most finitely many of the premises, and no such finite subset of the premises will be unsatisfiable by itself, or unsatisfiable with the claim that there are at most finitely many things.

Compactness for $\mathscr{L}_1$ Is $\mathscr{L}_1$ compact? This question can be answered in a number of ways. In this chapter, we show directly, just by consideration of the semantics of $\mathscr{L}_1$, that $\mathscr{L}_1$ is in fact compact. In chapter 7, after we consider a derivation system for $\mathscr{L}_1$, we can use the adequacy of that system to provide another more proof of compactness for $\mathscr{L}_1$(Corollary 87).

Compactness for $\mathscr{L}_1$ has the immediate consequence that if $\Gamma \vDash \phi$, then there is some finite set of premises Γ' such that $\Gamma' \vDash \phi$; every valid argument in $\mathscr{L}_1$ with

[5]Each premise entails all the earlier premises; and each premise by itself is true in some finite situation. So if we took only finitely many of these premises, they are collectively equivalent to their highest numbered member (k), and then all those finitely many premises will be true in a situation which has k or more things in it, for some finite k.

infinitely many premises has a corresponding valid argument with only finitely many premises. As suggested above in connection with the interpolation theorem, compactness is of interest in part because of the computational tractability of compact languages. We allow, for mathematical reasons, infinite sets of sentences to entail things. But we never *need* an infinite set of sentences to formulate an argument: any conclusion expressible in $\mathscr{L}_1$ that is validly entailed by infinitely many premises formulated in $\mathscr{L}_1$ is also validly entailed by some finite subset of those premises. We will return to this topic below when we discuss *decidability* (p. 96).

Definition 86 (Finite Satisfiability). We say that a set Γ is *finitely satisfiable* iff every finite subset of Γ is satisfiable (i.e., each has a model – each is consistent).

Theorem 60 (Compactness)
If Γ is finitely satisfiable then Γ is satisfiable.

This theorem can be proved a number of ways. It is a fairly quick consequence of the Completeness theorem for the natural deduction formalism for sentential logic, which we prove in chapter 6. In this chapter, we will prove it directly, in what is probably the most involved proof in this book. The proof comes in several stages, marked out below.

Note that the *converse* of Compactness – that satisfiability entails finite satisfiability – is trivial from the structural rules.

Proof of Compactness, Stage 1: Definition of Γ_is Suppose that Γ is finitely satisfiable. The set of sentence letters occurring in Γ is enumerable, so let its members be enumerated $s_1, s_2, \ldots$.

We define a sequence of supersets of Γ as follows:

$$\begin{aligned}
\Gamma_0 &= \Gamma \\
&\vdots \\
\Gamma_{n+1} &= \begin{cases} \Gamma_n \cup \{s_{n+1}\} & \text{if } \Gamma_n \cup \{s_{n+1}\} \text{ is finitely satisfiable;} \\ \Gamma_n \cup \{\neg s_{n+1}\} & \text{otherwise.} \end{cases}
\end{aligned}$$

Proof of Compactness, Stage 2: Γ_is are finitely satisfiable

Lemma 61
Each Γ_i is finitely satisfiable.

Proof. Base case: $\Gamma_0 = \Gamma$, which is finitely satisfiable by hypothesis.

Induction step: Suppose that each Γ_i, $i \leqslant n$, is finitely satisfiable. Now suppose Γ_{n+1} is not. Then, by definition of Γ_{n+1}, neither $\Gamma_n \cup \{s_{n+1}\}$ nor $\Gamma_n \cup \{\neg s_{n+1}\}$ is finitely satisfiable.

So there must be finite subsets Δ and Θ of Γ_n such that both $\Delta, s_{n+1} \vDash$ and $\Theta, \neg s_{n+1} \vDash$. By weakening, both $\Delta, \Theta, s_{n+1} \vDash$ and $\Delta, \Theta, \neg s_{n+1} \vDash$.

But $\Delta \cup \Theta$ is a finite subset of a finitely satisfiable set, so is consistent. In any structure $\mathscr{A}$ in which all the members of $\Delta \cup \Theta$ are true, either $[\![s_{n+1}]\!]_{\mathscr{A}} = T$ or $[\![\neg s_{n+1}]\!]_{\mathscr{A}} = T$. So either $\Delta, \Theta, s_{n+1} \nvDash$ or $\Delta, \Theta, \neg s_{n+1} \nvDash$. Contadiction; so Γ_{n+1} is finitely satisfiable. □

Proof of Compactness, Stage 3: Define a structure from the Γ_is For all s_i, define

$$\mathscr{G}(s_i) = \begin{cases} T & \text{if } s_i \in \Gamma_i; \\ F & \text{otherwise (i.e., if } \neg s_i \in \Gamma_i). \end{cases}$$

For any k, Γ_k is finitely satisfiable; since for all $i \leqslant k$, either $s_i \in \Gamma_k$ or $\neg s_i \in \Gamma_k$, the finite set of literals

$$S_k = \{s_i \in \Gamma_k : i \leqslant k\} \cup \{\neg s_i \in \Gamma_k : i \leqslant k\}$$

is a subset of Γ_k and is consistent.

By construction, for any k, and for all $\sigma \in S_k$, $[\![\sigma]\!]_{\mathscr{G}} = T$. Moreover, *every* structure $\mathscr{G}'$ which agrees with $\mathscr{G}$ on the sentence letters in S_k also assigns T to all members of S_k. (And obviously, by construction, only those structures which agree on these sentence letters can satisfy S_k.)

Proof of Compactness, Stage 4: $\mathscr{G}$ satisfies Γ Suppose $\mathscr{G}$ does not satisfy Γ. Then for some $\gamma \in \Gamma$, $[\![\gamma]\!]_{\mathscr{G}} = F$.

γ has finitely many sentence letters occurring in it; let s_k be the highest numbered. Since every $\sigma \in S_k$ is true in every structure which agrees with $\mathscr{G}$ on every s_i, $i \leqslant k$, γ is false in every such structure.

So $S_k \cup \{\gamma\}$ is unsatisfiable. But it is a finite subset of Γ_k, which is finitely satisfiable, so $S_k \cup \{\gamma\}$ satisfiable. Contradiction; so $\mathscr{G}$ must satisfy Γ. Γ is satisfiable if all of its finite subsets are.

5.6 An Alternative Proof of Compactness

In the proof of Compactness just given, we built up the structure that satisfies Γ by creating ever larger supersets of Γ – we added literals one-by-one in a way that preserves finite satisfiability, and then showed that the structure read off those literals satisfies Γ.

In this alternative proof of compactness, we build up the structure that satisfies Γ by creating a sequence of structures that converges ever closer to the desired structure, by getting more and more of the sentence letters needed to satisfy Γ right, until eventually every sentence letter has been dealt with.

n-Acceptable structures Suppose that Γ is a finitely satisfiable set of sentences, and $s_1, s_2, \ldots$ is an enumeration of the sentence letters occurring as constituents of sentences in Γ. Let $\mathbf{S}_n$ be the set consisting of the first n sentence letters in Γ under the enumeration, i.e., $\mathbf{S}_n = \{s_1, s_2, \ldots, s_n\}$. ($\mathbf{S}_0$ is the empty set.)

Call an $\mathscr{L}_1$-structure $\mathscr{B}$ *n-acceptable* iff, for each finite subset $\Delta \subseteq \Gamma$, there exists a structure which agrees with $\mathscr{B}$ on $\mathbf{S}_n$ which satisfies Δ. (Perhaps this is $\mathscr{B}$ itself, or perhaps $\mathscr{B}$ assigns something to a sentence letter s_{n+i} which prevents it from satisfying Δ.)

Define a series of $\mathscr{L}_1$-structures as follows:

- $\mathscr{A}_0$ is some arbitrarily chosen structure.
- If (i) $\mathscr{B}$ agrees with $\mathscr{A}_n$ on $\mathbf{S}_n$ and (ii) $\mathscr{B}$ assigns T to s_{n+1} and (iii) $\mathscr{B}$ is $(n+1)$-acceptable, then let $\mathscr{A}_{n+1}$ be $\mathscr{B}$; otherwise let $\mathscr{A}_{n+1}$ be any structure which agrees with $\mathscr{A}_n$ on $\mathbf{S}_n$ and which assigns F to s_{n+1}.

Each $\mathscr{A}_n$ is n-acceptable We now prove by induction on n that each $\mathscr{A}_n$ is n-acceptable (i.e, $\mathscr{A}_0$ is 0-acceptable, $\mathscr{A}_1$ is 1-acceptable, and so on).

Obviously $\mathscr{A}_0$ is 0-acceptable; to be 0-acceptable is for each finite subset of Γ to agree with $\mathscr{A}_0$ on the set of sentence letters in $\mathbf{S}_0$; since the latter is the empty set, *any* finite subset of Γ agrees with $\mathscr{A}_0$ on that set.

For the induction step, suppose there is some $\mathscr{A}_n$ which is n-acceptable, but $\mathscr{A}_{n+1}$ is not. Since by definition of the $\mathscr{A}_i$s, if there was any $(n+1)$-acceptable structure which agrees with $\mathscr{A}_n$ on $\mathbf{S}_n$ and which assigned T to s_{n+1}, $\mathscr{A}_{n+1}$ would be an example of such a structure. So $\mathscr{A}_{n+1}$ must assign F to s_{n+1}. But from the induction hypothesis that $\mathscr{A}_{n+1}$ is not n-acceptable, some finite set $\Delta \subset \Gamma$ is not satisfied by any structure which agrees with $\mathscr{A}_{n+1}$ on $\mathbf{S}_{n+1}$. So Δ cannot be satisfied by *any* structure which agrees with $\mathscr{A}_n$ on $\mathbf{S}_n$ (since it can't be satisfied by a variant of $\mathscr{A}_n$ which assigns T to s_{n+1}, or one that assigns F, and they are the only options), contrary to the assumption that $\mathscr{A}_n$ is n-acceptable. So $\mathscr{A}_{n+1}$ must in fact be $(n+1)$-acceptable.

Compactness Again Let $\mathscr{A}$ be a structure such that, for all i, $[\![s_i]\!]_{\mathscr{A}} = [\![s_i]\!]_{\mathscr{A}_i}$ (that is, for each i, $\mathscr{A}$ agrees with the i-th structure $\mathscr{A}_i$ on the value it assigns to the i-th sentence letter s_i).

Suppose $\mathscr{A}$ isn't a model for Γ. Then there is a $\phi \in \Gamma$ such that $[\![\phi]\!]_{\mathscr{A}} = F$. Let s_k be the highest numbered sentence letter occurring in ϕ. Then $\mathscr{A}_k$ must make ϕ true, since $\{\phi\}$ is a finite subset of Γ and $\mathscr{A}_k$ is k-acceptable. But then every structure which agrees with $\mathscr{A}_k$ on $\mathbf{S}_k$ will make ϕ true; and $\mathscr{A}$ is one such agreeing structure. Contradiction – $\mathscr{A}$ must satisfy Γ.

But since Γ was arbitrary, we've shown that if Γ is finitely satisfiable, then Γ is satisfiable. This is compactness.

5.7 Decidability

Effective Procedures An *effective procedure* for determining some query is an automatic and mechanical algorithm that terminates in a *finite* time with a correct 'yes' or 'no' answer.

A property is *decidable* iff there is an effective procedure which tests for it.

Turing Machines and Turing's Thesis The notion of an effective procedure has only been introduced informally here, but it can be made precise in a number of

different ways (Boolos *et al.*, 2007: chs. 1–8). In the 1930s, Turing introduced the notion of a *Turing machine*, which is in effect an extremely simplified computer (Turing, 1937). The simplicity of Turing machines establishes that anything they can do really is computable by an effective procedure, in the intuitive sense. What they primarily do is compute functions: when given a argument as input, a well-designed Turing machine will give a certain value as output, and the totality of all such input-output pairs will be a function.

Turing's thesis is the claim that any effectively computable function can be calculated by some Turing machine. This *prima facie* remarkable claim cannot be proved purely mathematically: it is a philosophical claim about the pre-theoretical notion of 'effectively computable'. But some evidence in its favour can be provided. First we might note that the class of functions calculable by Turing machines is the same class of functions calculable by another independent attempt to characterise the notion of an effective procedure, Church's theory of *recursive functions*, also developed in the 1930s. That these two quite different ways to make the informal notion of computability precise end up coinciding is some evidence for the success of both in capturing some robust notion. Second, no one has developed any example of a Turing-uncomputable function that is, intuitively, represented by an effective procedure. That a philosophical claim has survived 80 years of attempts to provide counterexamples is some evidence in its favour!

Finite Decidability Using Truth Tables

Theorem 62 (Decidability of Finite Validity)
If Γ *is a finite set of sentences of* $\mathscr{L}_1$*, then it is decidable whether* $\Gamma \vDash$*.*

Proof. Suppose Γ is a finite set of $\mathscr{L}_1$ sentences. Each $\gamma \in \Gamma$ is only finitely long, so there are n sentence letters occurring in Γ, for some finite n. The following describes an effective procedure that can be followed to determine whether Γ is unsatisfiable.

> Construct a truth-table with 2^n lines such that each line corresponds to some structure assigning truth values to each sentence letter in Γ; this can be done mechanically in a finite time. For each $\gamma \in \Gamma$, determine its truth value in each row; this can be done mechanically in a finite time for each γ in each row, and there are finitely many

rows, so the truth value of each γ in every class of structures that agree on sentence letters in Γ can be determined mechanically in a finite time. Since Γ is finite, we can determine the truth value of each sentence in Γ in every structure in a finite time.

Check each row of the truth table to see if all sentences in Γ have been assigned T: If they all have, in some row, then stop, and answer 'No' to the query, 'Is $\Gamma \vDash$?'. If no row has all $\gamma \in \Gamma$ assigned T, then answer 'Yes'. There are only finitely many rows, so this final check can be done in a finite time. All steps can be finitely mechanised, and there are finitely many steps: inconsistency is decidable. □

This has the immediate consequence that it is decidable whether $\Gamma \vDash \phi$ (just check whether $\Gamma, \neg\phi \vDash$ by Theorem 26), and thus that *validity is decidable*.

Definition 87 (Recursivity). A set $N \subseteq \mathbb{N}$ is *recursive* iff there is an effective procedure such that, given an arbitrary number $n \in \mathbb{N}$, tells us whether $n \in N$ or $n \notin N$.

A recursive set has an associated decision procedure that sorts every candidate number into members and non-members. Example: the set of prime numbers is recursive. Given a number n, successively try to divide n by all numbers between 2 and $n/2$. This task is finite: if you succeed in finding a divisor, n isn't prime, and if you don't, it is prime. Either way, you find out in a finite time.

Because of the relationship between $\mathscr{L}_1$ sentences and numbers established by the code in Lemma 22, we can by courtesy extend the notion of a recursive set of $\mathscr{L}_1$ sentences in the obvious way: a set of $\mathscr{L}_1$ sentences is recursive iff there is an effective procedure which decides whether an arbitrary $\mathscr{L}_1$ sentence is a member of it or not. Theorem 62 shows that the set of $\mathscr{L}_1$ contradictions is recursive. Any $\mathscr{L}_1$ sentence is equivalent to a conjunction, by the CNF theorem. An contradiction is thus equivalent to some contradictory conjunction $\delta_1 \wedge \ldots \wedge \delta_n$, where each δ_i is a disjunction of literals. But a conjunction $\delta_1 \wedge \ldots \wedge \delta_n$, is a contradiction iff the finite set $\{\delta_1, \ldots, \delta_n\}$ is unsatisfiable. Since whether a finite set is unsatisfiable is decidable, and whether two finite sentences are logically equivalent is also decidable (exercise), for any sentence it is decidable whether

it is a contradiction or not. So the set of contradictions is recursive. A similar argument shows that the set of all $\mathscr{L}_1$ tautologies is recursive.

Recursive Enumerability A query is *positively decidable* iff an effective procedure will terminate and correctly answer 'Yes' in a finite time. A query is *negatively decidable* iff an effective procedure will terminate and correctly answer 'No' in a finite time. It is decidable iff it is both positively and negatively decidable.

Definition 88 (Recursive Enumerability). A set S is *recursively enumerable* iff there is an effective procedure that successively lists all and only the members of S. (More formally, if S is the range of a computable function whose domain is a subset of $\mathbb{N}$ – this is just like the definition of a enumeration in Definition 45 except that now we require the enumeration function to be computable.)

If S is infinite, this procedure will never halt. Nevertheless, if S is recursively enumerable, every member of S will eventually be listed after some finite time from the beginning of the procedure.

Recursive and Recursively Enumerable All recursive sets are recursively enumerable. At stage n, apply the procedure to check whether $n \in N$ or not; if yes, print 'n'; if no, print nothing; then apply the same procedure to $n + 1$. Begin at $n = 1$, and this will successively enumerate the members of N.

But not all recursively enumerable sets are recursive. If N is recursively enumerable, whether $n \in N$ is positively decidable: for we just have to set the N-enumerating procedure in motion, watch its output, and if n comes up, we stop the procedure and announce that $n \in N$. But if in fact $n \notin N$, we will fruitlessly watch the output of the N-enumerating procedure forever, never sure whether the next output will be n or not. (In reality, it will never come up, but we don't know that.)

For a concrete example, consider the set of *prime gaps*, where a prime gap is a number n such that it is the difference between successive primes (prime numbers between which there are no other prime numbers: so 7 and 13 are successive primes with a gap of 4). The set of prime gaps is $\{n : \exists i(n = p_{i+1} - p_i)\}$. This set is not recursive – given an arbitrary number n that is not a prime gap, no effective procedure will determine that. We can effectively generate primes, and

once we have generated them, we can check their gaps – so if n is a prime gap, we'll eventually discover that after only a finite time. But until we generate all the infinitely many prime numbers, we cannot survey the whole and see that our unluckily chosen number number isn't among the prime gaps.

We can extend the notion of recursive enumerability to sentences of $\mathscr{L}_1$ too, and finite sets of sentences of $\mathscr{L}_1$. Lemma 22 gives us a representation of $\mathscr{L}_1$ sentences as numbers; we can then use that same coding to represent finite sequences of numbers as numbers too. Hence we may encode a finite set of $\mathscr{L}_1$ sentences by a number; and we may then say that a set of $\mathscr{L}_1$ sentences, or a set of sets of $\mathscr{L}_1$ sentences, is recursively enumerable if the corresponding sets of code numbers are recursively enumerable.

Positive Decidability of Unsatisfiability It is not possible to effectively determine the members of every set of $\mathscr{L}_1$ sentences. For the number of effective procedures is countable, given the plausible assumption that an effective procedure can be specified by a finite recipe. (See the exercises for more on uncomputable functions.) But the number of infinite sets of $\mathscr{L}_1$ sentences is uncountable (being the powerset of the countable set of all $\mathscr{L}_1$ sentences). So not every enumerable set of sentences Γ can be effectively enumerated. But those whose members can be effectively listed – the recursively enumerable sets – can also be effectively checked for unsatisfiability.

Theorem 63 (Positive Decidability of Unsatisfiability)
If Γ is a recursively enumerable set of $\mathscr{L}_1$ sentences, then if $\Gamma \vDash$, there is an effective procedure that will demonstrate its unsatisfiability.

Proof. Let the sentences in Γ be enumerated $\gamma_1, \ldots, \gamma_n, \ldots$ by some recursive function that enumerates it.

At each stage i, an initial subset $G_i = \{\gamma_1, \ldots, \gamma_i\}$ will have been listed by the enumeration procedure. Before moving on to produce γ_{i+1}, apply the procedure from Theorem 62 to G_i; if G_i is unsatisfiable, then halt and say that Γ is unsatisfiable. If G_i is satisfiable, then move on to generate γ_{i+1} and repeat the process.

If Γ is unsatisfiable, then by Compactness there is some j such that G_j is finite and unsatisfiable. Any finite unsatisfiable subset of Γ will have its members in some finite initial subset of Γ. (Obviously G_k for $k > j$ will also be unsatisfiable.)

So if Γ is unsatisfiable, this process will eventually detect that by coming across some finite initial subset of Γ which is unsatisfiable.

If Γ is satisfiable, however, this procedure will never halt: it will keep checking larger and larger initial subsets of Γ for unsatisfiability, never finding any of them to be unsatisfiable, but also never running out of new larger candidates to check.□

This procedure is effective but resource inefficient – depending on the details of how Γ is enumerated, a more optimised algorithm could be designed. But the limitation that it cannot check for satisfiability is not a practical limitation, but an in principle one. Thus whether a set is unsatisfiable is positively decidable (if we apply it to an unsatisfiable set, our procedure will tell us that it is unsatisfiable), but not negatively decidable (if we apply it to a satisfiable set, our procedure and any similar procedure will not halt in its fruitless search for an unsatisfiable subset).

Exercises

1. (a) Show that $(\phi \wedge \psi) \wedge \chi)$ is logically equivalent to $\dashv\models (\phi \wedge (\psi \wedge \chi))$.
   (b) Show that $(\phi \wedge \psi)$ is logically equivalent to $(\psi \wedge \phi)$.
   (c) What significance do the foregoing results have for the truth-function that '$\wedge$' expresses?
2. (a) Show that if there are n sentence letters in S, there are 2^n sentences of the form $\mathfrak{c}_{\mathscr{A}}$ defined in the proof of the DNF theorem.
   (b) Explain clearly why, in the proof of the DNF theorem, the sentence $\mathfrak{d}$ there constructed expresses the truth function f.
3. Prove, by an argument analogous to the DNF theorem, this claim:

   Theorem 64 (CNF)
   Every truth function is expressed by a sentence in Conjunctive Normal Form.

4. Show that all truth-functions which can be expressed using only $\rightarrow$ and $\wedge$ are positive. (Prove by induction on complexity of sentences.)
5. The proof of theorem 50 was only sketched. Fill in these two crucial gaps.
   (a) Show that $\phi \vee \psi$ is logically equivalent to $(\psi \rightarrow \phi) \rightarrow \phi$;
   (b) Show (by induction on complexity of sentences, and using the above result) that each conjunct of a CNF sentence will be equivalent to either (i) an arrow sentence *without* negation, or (ii) a conjunct of the form $\bigvee_i \neg s_i$.

(c) Show that a necessary and sufficient condition for a sentence ϕ to be logically equivalent to a sentence involving just the truth functional connectives $\rightarrow$ and $\wedge$ is that the sentence has the value T in any structure that assigns the value T to each sentence letter in ϕ.

6. (a) Show that $\{\vee, \neg\}$ is expressively adequate.

(b) Is $\{\rightarrow, \neg\}$ expressively adequate?

(c) Is $\{\leftrightarrow, \wedge, \rightarrow, \vee\}$ expressively adequate?

(d) Is any connective in $\mathscr{L}_1$ expressively adequate by itself?

(e) Prove that $\downarrow$ is expressively adequate.

(f) How many 2-place truth-functional connectives are expressively adequate by themselves?

(g) Consider the 0-place connective $\bot$ that expresses the constant 0-place function $f : \varnothing \mapsto F$. Is this expressively adequate, or can you add a connective to get an expressively adequate set (where the added connective is not itself expressively adequate)?

7. (a) Consider the self-dual set of connectives $\{\rightarrow, \rightarrow^\star\}$. Is this set of connectives expressively adequate? Is self-duality either necessary or sufficient for expressive adequacy of a set of connectives?

(b) Show that if a two-place truth functional connective $\oplus$ is self-dual, then the function that expresses it, $f_\oplus$, must be such that $f_\oplus(T, F) \neq f_\oplus(F, T)$ and $f_\oplus(F, F) \neq f_\oplus(T, T)$. Make use of this result, establish how many self-dual two-place truth functors there are.

(c) Let ϕ and ψ be sentences of $\mathscr{L}_{\neg,\wedge,\vee}$. We say that a sentence ϕ is *dualable* iff for all $\mathscr{A}$, $(\llbracket\phi\rrbracket_{\mathscr{A}^\star})^\star = \llbracket\phi^\star\rrbracket_{\mathscr{A}}$.

i. Prove that if ϕ and ψ are dualable, so too are $\neg\phi$, $(\phi \wedge \psi)$, and $(\phi \vee \psi)$.

ii. Prove using this result that all sentences of $\mathscr{L}_{\neg,\wedge,\vee}$ are dualable.

8. Let $\mathscr{L}_{\uparrow,\downarrow}$ be the language which has $\uparrow$ and $\downarrow$ as its only connectives.

(a) For any ϕ in $\mathscr{L}_{\uparrow,\downarrow}$, and any structure $\mathscr{A}$, show that $\llbracket\phi\rrbracket_{\mathscr{A}} = \mathbf{n}(\llbracket\phi^\star\rrbracket_{\mathscr{A}^\star})$.

(b) For any $\mathscr{L}_{\uparrow,\downarrow}$-sentence ϕ, let $\phi^\star$ be the sentence which results from replacing every connective in ϕ by its dual, and let $\overline{\phi}$ be the result of substituting $(P_i \uparrow P_i)$ for every sentence letter P_i occurring in ϕ.

i. Prove that for every $\mathscr{L}_{\uparrow,\downarrow}$ sentence ϕ, $\phi^\star$ is logically equivalent to $(\overline{\phi} \uparrow \overline{\phi})$.

ii. Prove that if ϕ and ψ are $\mathscr{L}_{\uparrow,\downarrow}$ sentences, $\phi \vDash \psi$ iff $\psi^\star \vDash \phi^\star$.

9. Find an interpolant for the following sequents. Be sure in each case to give the simplest interpolant (i.e., find the most elegant sentence that is equivalent to your chosen interpolant).

(a) $((Q \vee P) \to R) \vDash ((P_1 \wedge \neg R) \to \neg Q)$;

(b) $(\neg(P \vee Q) \wedge (P \leftrightarrow R)) \vDash ((R \to P) \wedge \neg(P_1 \wedge R))$;

(c) $((Q_2 \leftrightarrow Q) \wedge \neg((R \to \neg P_1) \vee \neg(P \to Q))) \vDash (R_1 \to (P_2 \to (\neg P \vee Q)))$;

10. (a) Let Γ be a possibly infinite set of sentences of $\mathscr{L}_1$ such that $\Gamma \vDash$. Show that there is a finite disjunction, δ, each disjunct of which is the negation of a sentence in Γ, and such that $\vDash \delta$.

(b) Consider the following relation holding between sets of sentences:

> Where Γ and Δ are any sets of sentences, $\Gamma \vDash^\star \Delta$ is correct iff every structure which satisfies *every* $\gamma \in \Gamma$ is also one which satisfies *at least one* $\delta \in \Delta$.

Show that if $\Gamma \vDash^\star \Delta$ and neither is empty (though either or both may be infinite), there is a finite conjunction of $\mathscr{L}_1$ sentences in Γ,

$$\Phi = (\phi_1 \wedge \ldots \wedge \phi_m),$$

and a finite disjunction of $\mathscr{L}_1$ sentences in Δ,

$$\Psi = (\psi_1 \vee \ldots \vee \psi_n),$$

such that $\Phi \vDash \Psi$. (You may assume the Compactness theorem.)

(c) A set of sentences of English is *compossible* just in case it is possible for them all to be true together. An analogue for the compactness theorem in English would be: for every infinite set E of English sentences not all compossible, there is a finite subset of E whose members are not all compossible.

i. Show that this analogue of compactness fails for English.

ii. What does this show about translations from English into $\mathscr{L}_1$?

11. Intuitively, any effective procedure must be able to be written down by a finite string of sentences in some language – say, English.

(a) Give an argument that the set of effective procedures is countable.

(b) Let a *recipe* be any finite set of English sentences. Consider the set E of recipes. (It is obvious that the set of effective procedures in English which compute one-place functions whose domain is (a subset of) the natural numbers will be a subset of E.) This set is countable; let f_n be the function – if there is one – computed by the n-th recipe in E under some enumeration of E. Define

$$d(n) = \begin{cases} 0 & \text{if } f_n(n) \text{ is defined and equal to 1;} \\ 1 & \text{otherwise} \end{cases}$$

i. Show that there is no effective procedure for computing d.

ii. Show that there is no effective procedure for deciding if a recipe is an effective procedure, and therefore that while the set of recipes E can be effectively produced, some subsets of E (in particular, the one corresponding to the set of effective procedures) cannot be effectively produced.

(c) Give an argument for the claim that there is an effective procedure for determining whether two $\mathscr{L}_1$ sentences are logically equivalent.

Answers to selected exercises on page 243.

Part III

Sentential Logic: Derivations

Chapter 6

Tableau Derivations in $\mathscr{L}_1$

6.1 Derivations

The key notion in logic is, as we've said earlier (section 4.3), *consequence*: the notion of some sentences following from another. So far, we've understood consequence in terms of *entailment*, so that ϕ is a consequence of Γ just in case every structure in which all the sentences in Γ are true (in which Γ is satisfied), is also a structure in which ϕ is true.

Consequence as a Semantic Notion There is, however, something a bit strange about understanding consequence solely in this way. On this account, what underlies whether some sentences have a certain consequence ϕ are facts about the class of structures in which those sentences are true, in particular, whether that class is a subset of some further class which corresponds to the class of structures in which ϕ is true. Rather than reasoning from Γ to ϕ, we reason about the structures in which Γ and ϕ are true. And this seems to be a detour. Compare these two English arguments:

(4)	If it's raining, I won't ride my bike.	(4′)	'If it's raining, I won't ride my bike' is true.
(5)	It's raining.	(5′)	'It's raining' is true.
(6)	Therefore, I won't ride my bike.	(6′)	Therefore, 'I won't ride my bike' is true.

The second argument does the same job as the first, when supplemented with this principle: For any sentence S,

(T-schema) $\ulcorner S \urcorner$ is true iff S.

But it is needlessly complex, detouring through the metalanguage when the first object-language only argument does the same job. English is its own metalanguage, so the detour isn't so noticeable. But in $\mathscr{L}_1$, where the metalanguage is English, and therefore quite distinct from $\mathscr{L}_1$ itself, the results can be striking. The standard technique for evaluating an argument in $\mathscr{L}_1$ – truth tables – produces results that require a great deal more to understand them than is required to understand $\mathscr{L}_1$. $\mathscr{L}_1$ is a very simple language, which cannot express much. To understand a truth table, however, one has to understand the notion of truth; of a structure; of a sentence having a truth value in a structure; and all the set theoretic and mathematical background to these notions. This shows already that the *activities* of deriving (6) and deriving (6′) may be quite different, needing different conceptual resources. Young children, for example, might well be able to run through the object language reasoning without being able to reason in the metalanguage.

Formal Argument The goodness of the object language argument (4)–(6) doesn't consist in anything more than the fact that its first premise is a conditional, the antecedent of which is the second premise and the consequent of which is the conclusion. We don't need to know anything more than this fact about the structure of the argument, and the rules governing conditional implication, to establish that this object language argument is good. The rule governing conditional implication in question is this:

(*Modus Ponens*) If claims ⌜ϕ⌝ and ⌜If ϕ then ψ⌝ occur in the course of an argument, you may subsequently derive ⌜ψ⌝.

You don't need to know what ϕ or ψ mean to obey this rule – you don't even need to know what 'if — then —' means. You just need to be able to follow the rule. Maybe it is purely formal features of argumentation like this that allow young children to follow and produce chains of reasoning in ignorance of semantic notions. Of course, in cases like the above the metalanguage argument complements the object language argument, by showing the formal argument to have a legitimate form, one where the premises really do entail the conclusion. So it is not as if the metalanguage argument is otiose. What it does seem to be is unduly complex for carrying out an inferential task that one might more directly deploy the object language argument to accomplish. At present, we have no choice, since we have no tools for reasoning directly between $\mathscr{L}_1$ sentences rather than detouring via the semantics. So perhaps we ought to remedy that lack.

Philosophical Considerations Some have thought that it is philosophically preferable to develop systems of formal argumentation in a language, rather than having to always evaluate those arguments using its semantics. Some have been motivated by scepticism about truth – perhaps motivated by the Liar paradox ('This sentence is not true'). The technical notion of *satisfaction in an $\mathscr{L}_1$-structure* which is at the heart of semantics for $\mathscr{L}_1$ doesn't seem to be susceptible to the same sort of paradox, not least because we cannot formulate self-referential sentences in $\mathscr{L}_1$. But some have thought that if $\mathscr{L}_1$ is to be a paradigm for evaluating natural language arguments, then we ought to avoid beginning with semantic notions in formal languages to avoid being forced to begin with semantic notions when we turn to natural language.

Other considerations in favour of developing methods of object language formal argument come from thinking about effective computation. One nice feature of formal arguments like *modus ponens* is that it is fairly clear that they are effectively implementable. It might not be easy to program a computer to evaluate whether every model in which $P \to Q$ and P are true is also one in which Q is true. But it is easy to program a computer to add Q to a body of claims if P and $P \to Q$ are already among those claims. Machines are good at performing mechanical syn-

tactic manipulations, and rather worse at evaluating expressions for their meaning. Since it is desirable to be able to carry out formal reasoning in $\mathscr{L}_1$ effectively, we need to develop techniques that a 'mere' computer might be able to use.

Some even hold out the hope that these formal inference rules might be all there is to intelligent behaviour. A system following a complicated enough set of formal rules might produce outputs indistinguishable from the intelligent behaviour produced by creatures like us. Not only would this hold out the promise of 'artificial intelligence', it might lead to a dissolution of philosophical puzzles about our own conscious reasoning behaviour. Maybe for all we know, we are just complicated formal rule following systems.

Finally – more mundane, but perhaps most significant historically – it is often the case that the development of formal languages runs ahead of their semantics. We might begin by trying to translate some fragment of natural language into a formal language, introducing some bits of formal language to represent certain natural language expressions. We might characterise the behaviour of those formal language constructions by some rules, hoping thereby to capture some significant part of the natural language expressions with which we began. We can use these rules to construct certain arguments, which may mirror in their structure intuitively acceptable natural language arguments. This might give us some confidence that we've captured something interesting in the natural language. But we don't have anything yet that looks like a full theory of meaning for this language – we don't have enough to construct a relation of entailment, for example. But this wouldn't prevent us from going on and developing various rival formal systems to capture various aspects of natural language. We might have some informal understanding of those languages in virtue of the fact that they were constructed to mimic natural language. But we might not be confident that every formal argument that our system endorses corresponds to a real entailment, nor confident that our system can capture every real entailment. The case of sentential logic is not typical here. Rather more typical is the case of modal logic – the logic of 'necessarily' and 'possibly'. In this case various inequivalent and rival systems of formal argument were set up as early as the 1930s, but the semantics for those languages which allowed a precise account of modal entailment did not come along until the 1960s. It would have been absurd to suggest that people shouldn't have been investigating the formal structure of modal arguments for those 30 years.

The informal gloss on their syntax was enough for people to pursue interesting formal results in their system, even if a complete semantics is nevertheless the ideal.

Derivations A *derivation* will be a certain structured collection of $\mathscr{L}_1$ sentences, and a *derivation system* will be a collection of rules that dictate how to create a derivation. (Many texts use the word 'proof' for what I am calling derivations, and the branch of logic known as 'proof theory' is in fact the mathematical theory of formal derivations.)

We will use derivations to implement argument in $\mathscr{L}_1$ directly, without needing to appeal to semantic notions like truth and satisfaction. However, a good derivation system for $\mathscr{L}_1$ will correspond to its semantics, in the sense that every acceptable derivation of ϕ from Γ will correspond to an entailment of ϕ by Γ; and every entailment will correspond to an acceptable derivation. Proving that the derivation systems we consider are adequate to the semantics for $\mathscr{L}_1$ in this sense is the main task of chapter 7 and chapter 9.

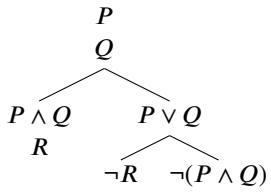
There are many derivation systems, giving rise to many different kinds of derivation. In this book, we look at two in detail: *truth trees* or *tableaux*, in the present chapter; and *natural deduction* in chapter 8. We will also briefly consider a third derivation system in chapter 9, an *axiomatic* system.

6.2 Trees

Before introducing our derivation system, let me introduce the idea of a tree. Informally, a tree is a collection of nodes, organised into branches, and all stemming from a single root node. (I give the precise definition, and show that our trees are indeed trees, in chapter 7.) We are only going to be concerned here with trees whose nodes are $\mathscr{L}_1$ sentences. I begin by defining an $\mathscr{L}_1$-branch:

Definition 89 ($\mathscr{L}_1$-Branch). An *$\mathscr{L}_1$-branch* is a finite ordered sequence of $\mathscr{L}_1$ sentences.

An $\mathscr{L}_1$-tree will be made up of $\mathscr{L}_1$-branches. Ordinarily, we assume that branches begin at the point of divergence from the parent branch or trunk. But $\mathscr{L}_1$-branches will not: they continue all the way to the root.



$B_1 = \langle P, Q, P \wedge Q, R \rangle$;
$B_2 = \langle P, Q, P \vee Q, \neg R \rangle$;
$B_3 = \langle P, Q, P \vee Q, \neg(P \wedge Q) \rangle$.

Figure 6.1: An $\mathscr{L}_1$-tree and its associated $\mathscr{L}_1$-branches

Definition 90 ($\mathscr{L}_1$-Tree). An $\mathscr{L}_1$-*tree* $\mathbf{T}$ is a set of $\mathscr{L}_1$-branches such that there is some sequence of $\mathscr{L}_1$ sentences $\langle \phi_1, \ldots, \phi_n \rangle$ that is the initial subsequence of every branch $B \in \mathbf{T}$. (This common initial subsequence is called the *trunk*, and the initial node common to all branches is called the *root*.)

$\mathscr{L}_1$-branches on $\mathscr{L}_1$-trees thus overlap one another, by having initial subsequences in common. Two $\mathscr{L}_1$-branches *diverge* at a position i if they share an initial sequence up to position i, but differ at position i at least. Because two $\mathscr{L}_1$-branches are identical iff they have the same sentences in the same positions, there cannot be an $\mathscr{L}_1$-tree which diverges and has the same sentences on each branch after the divergence.

Drawing $\mathscr{L}_1$-Trees Where two $\mathscr{L}_1$-branches overlap, we do not need to represent the sentences in the overlap twice, so when we draw an $\mathscr{L}_1$-tree, we will draw any sentences (including the root node) on which all $\mathscr{L}_1$-branches overlap first, then when there is a divergence between $\mathscr{L}_1$-branches we will split the $\mathscr{L}_1$-tree, drawing then that subset of $\mathscr{L}_1$-branches which agree on one path of the divergence, and then that subset of $\mathscr{L}_1$-branches which agrees on the other path, and repeat until we have drawn all the nodes of the $\mathscr{L}_1$-branches. (Assuming the $\mathscr{L}_1$-tree is finite, every $\mathscr{L}_1$-branch terminates in a final node, known as a *leaf*.) An $\mathscr{L}_1$-tree meeting this definition and represented as suggested is pictured in Figure 6.1.

It is useful to note that the $\mathscr{L}_1$-tree pictured contains another $\mathscr{L}_1$-tree as a proper part. For the $\mathscr{L}_1$-branches $C_1 = \langle P \vee Q, \neg R \rangle$ and $C_2 = \langle P \vee Q, \neg(P \wedge Q) \rangle$ together meet the definition of an $\mathscr{L}_1$-tree with root node $P \vee Q$. We may formally

define this notion.

Definition 91 (Occurs Within). An $\mathscr{L}_1$-tree $\mathbf{T}$ *occurs within* an $\mathscr{L}_1$-tree $\mathbf{U}$ iff there exists some finite sequence σ such that (i) prefixing every $\mathscr{L}_1$-branch in $\mathbf{T}$ with σ yields an $\mathscr{L}_1$-branch in $\mathbf{U}$, and (ii) every $\mathscr{L}_1$-branch in $\mathbf{U}$ which begins with σ is the result of prefixing some $\mathscr{L}_1$-branch in $\mathbf{T}$ with σ.[1]

I will from now on often drop the qualification '$\mathscr{L}_1$-tree' and simply talk of 'trees'. When it matters for the detail of some argument, I will keep the terminological distinction intact.

6.3 Tableaux

We now turn to our promised derivation system. I will informally outline the basic idea, before giving a formal definition. The next section will give some worked examples.

The idea of tableau is to show that we can give a mechanical and automatic test of unsatisfiability of a set of sentences for a language – a test that can be applied even if we are in total ignorance of the intended semantic interpretation of the language. That will also be a test for validity of an associated argument, by Theorem 26. So tableaux is a formal representation in the object language of the proof technique of *reductio*.

An ordinary *reductio* involves some judgement about how to pursue the contradiction required to show the falsity of the *reductio* assumption. In a formal derivation, we don't want to have to involve judgement, since judgement will depend on understanding the contents of the materials on which we are operating. The key observation is this: when some $\mathscr{L}_1$ sentences are unsatisfiable, that is because there is no consistent way of assigning T and F to the sentence letters involved in those sentences to generate a structure in which each of the initial sentences is true. So if we can mechanically extract from a set of sentences its consequences for literals, we can then mechanically check whether there is a structure which makes all the unnegated literals true and negated literals false.

[1] Alternatively, for each sequence σ_i, let $\mathbf{\Sigma}_i$ be the subset of $\mathbf{U}$ containing all $\mathscr{L}_1$-branches begininng with σ_i. $\mathbf{T}$ occurs within $\mathbf{U}$ iff there exists some σ_j such that $\mathbf{T}$ consists of all final subsequences of $\mathscr{L}_1$-branches occuring in $\mathbf{\Sigma}_j$ obtained by removing the initial subsequence σ_j.

If there is not, that must be because a sentence letter and its negation are both consequences of assuming the original set to be satisfiable.

Tableaux Formally Defined That is the guiding idea. Here is the formal definition implementing it.

Definition 92 (Tableaux). A *tableau* (plural *tableaux*) is an $\mathscr{L}_1$-tree $\mathbf{T}$ such that every sentence ϕ on a node of $\mathbf{T}$ and any $\mathscr{L}_1$-branch $B \in \mathbf{T}$ meets one of the following conditions:

1. ϕ is a member of an initial subsequence of every $\mathscr{L}_1$-branch in $\mathbf{T}$; or
2. B has of one of these forms:
   (a) $\langle \rho, \ldots, \neg\neg\phi, \ldots, \phi, \ldots \rangle$;
   (b) $\langle \rho, \ldots, (\phi \wedge \psi), \ldots, \phi, \psi, \ldots \rangle$;
   (c) $\langle \rho, \ldots, \neg(\phi \vee \psi), \ldots, \neg\phi, \neg\psi, \ldots \rangle$;
   (d) $\langle \rho, \ldots, \neg(\phi \rightarrow \psi), \ldots, \phi, \neg\psi, \ldots \rangle$; or
3. Any pair of $\mathscr{L}_1$-branches B_1, B_2 which share an initial subsequence $\Sigma = \langle \rho, \ldots, \delta, \ldots \rangle$ on which some sentence δ appears, have one of these forms (remember that '$\frown$' is the concatentation operator on sequences):
   (a) $B_1 = \Sigma \frown \langle \phi, \ldots \rangle$, $B_2 = \Sigma \frown \langle \psi, \ldots \rangle$, and $\delta = (\phi \vee \psi)$;
   (b) $B_1 = \Sigma \frown \langle \neg\phi, \ldots \rangle$, $B_2 = \Sigma \frown \langle \neg\psi, \ldots \rangle$, and $\delta = \neg(\phi \wedge \psi)$;
   (c) $B_1 = \Sigma \frown \langle \neg\phi, \ldots \rangle$, $B_2 = \Sigma \frown \langle \psi, \ldots \rangle$, and $\delta = (\phi \rightarrow \psi)$;
   (d) $B_1 = \Sigma \frown \langle \phi, \psi \ldots \rangle$, $B_2 = \Sigma \frown \langle \neg\phi, \neg\psi, \ldots \rangle$, and $\delta = (\phi \leftrightarrow \psi)$;
   (e) $B_1 = \Sigma \frown \langle \phi, \neg\psi, \ldots \rangle$, $B_2 = \Sigma \frown \langle \neg\phi, \psi, \ldots \rangle$, and $\delta = \neg(\phi \leftrightarrow \psi)$.

Definition 93 (Generation of a tableau). A finite set of sentences Γ *generates* an $\mathscr{L}_1$-tableau $\mathbf{T}$ iff there is some sequence S such that each $\gamma \in \Gamma$ occurs once in S, no sentence not in Γ occurs in S, and S is a common initial subsequence of every $\mathscr{L}_1$-branch on $\mathbf{T}$ (i.e., is the trunk).

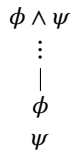
The same set of sentences can generate many tableaux, by taking the members of the set in different sequence, and applying the tableaux rules in different order. (See the two tableaux pictured in Figure 6.4, both generated by $\{(P \wedge Q), (\neg P \vee \neg Q)\}$.) And the same tableau can be generated by more than one set of sentences. Consider this one-branched tableaux: $\{\langle (P \wedge Q), P, Q\rangle\}$. It could have been generated from $\{(P \wedge Q)\}$ in accordance with the conjunction rules; or by $\{(P \wedge Q), P, Q\}$ in accordance with the condition on initial subsequences.

Pictorial Representation of Tableaux This precise definition is a bit hard to grasp, admittedly. It may be easier if we consider the tree structure pictorially. Then Definition 92 amounts to saying: If we have an finite set of sentences Γ, then construct an $\mathscr{L}_1$-tree generated by Γ such that every node on every branch subsequent to the shared trunk has been inscribed in accordance with the tableau rules pictured in Figure 6.2. You may read these rules as follows (in accordance with Definition 92):

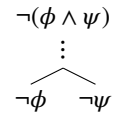
- The *list* rules in Figures 6.2a, 6.2d, 6.2f and 6.2i say: if a sentence of the form at the top of the rule occurs somewhere on a branch, then the sentence(s) at the bottom of the rule occur lower on the branch;
- The *branch* rules (all other rules in Figure 6.2) say: if a sentence of the form at the top of the rule occurs somewhere on on two branches which share an initial subsequence, then the sentence(s) on the left at the bottom of the rule are the first sentence(s) on one branch after that shared initial subsequence, and the sentence(s) on the right are the first sentence(s) on the other branch after that shared initial subsequence.

Here's the idea. Suppose we begin with the set $\{P \wedge Q, Q \rightarrow R, \neg(P \wedge R\}$. Inscribe this set in a sequence $S = \langle P \wedge Q, Q \rightarrow R, \neg(P \wedge R)\rangle$ – this is not the only way to do it, but it doesn't matter which way. The construction can be seen in Figure 6.3. We begin by inscribing the initial generating sequence in 6.3a. We apply the conjunction rule from Figure 6.2a in 6.3b, and then apply the conditional rule from Figure 6.2e in 6.3c.

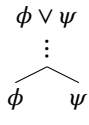
Finished Tableaux The tableau in Figure 6.3 satisfies Definition 92. But there is more we could do – for we could apply the negated conditional rule to $\neg(P \wedge R)$.



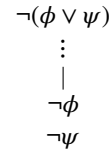
(a) Conjunction



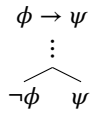
(b) Negated Conjunction



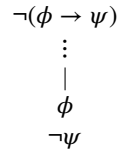
(c) Disjunction



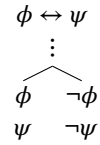
(d) Negated disjunction



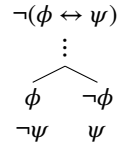
(e) Conditional



(f) Negated Conditional



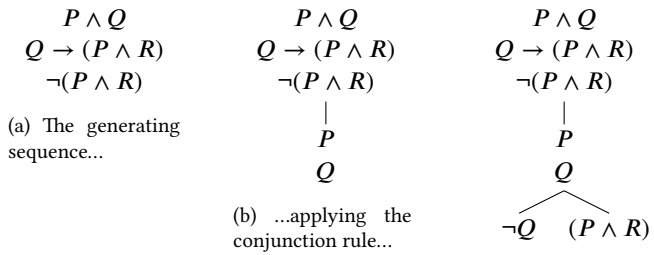
(g) Biconditional



(h) Negated Biconditional

(i) Double Negation

Figure 6.2: Sentential Tableau Rules



(a) The generating sequence...

(b) ...applying the conjunction rule...

(c) ...appling the conditional rule...

Figure 6.3: Steps to constructing a Tableau

When there is more that we could do to extend a tableau, we will say that the tableau is *unfinished*. Let us make this precise.

Definition 94 (Pruning Tips). A branch B results from *pruning* a branch B' iff B is a (proper or improper) initial subsequence of B'. If $B' = B \frown T$, we say that T is the *tip* of B' with respect to B.

Definition 95 (Extends). A tableau $\mathbf{T}'$ *extends* $\mathbf{T}$ iff $\mathbf{T}$ is a tableau and it results from pruning every branch in $\mathbf{T}'$.

Alternatively, a tableau extends another iff it results from that tableau by some number of applications of the tableau rules to some or all of its branches.

Definition 96 (Finished). A tableau $\mathbf{T}$ is *finished* iff any branch B' on any tableau $\mathbf{T}'$ that extends B on $\mathbf{T}$ is such that the tip of B' with respect to B contains only sentences which are already on B.

A tableau is finished not when we run out of rules to apply – we never will, since we can apply them repeatedly and keep extending any branch – but rather when once we start repeating ourselves with the application of the tableau rules. We see that repetition when any extension of a tableau, which comes by adding new sequences as the tips of old branches, ends up with those tips containing only sentences already appearing on the old branches.

Can Tableaux be Finished? We should reassure ourselves that tableaux can be finished – after all, it is a requirement on tableaux that they consist of finite branches, and we don't (yet) have a proof that we can reach the stage of repeating ourselves finitely far from the initial subsequence. So let us reassure ourselves in this respect.

Lemma 65 (Tableau Rules are Complexity-Decreasing)
If a tableau rule can be applied to a sentence ϕ, any sentences on the tableau licensed by that rule are of lower complexity than ϕ.

Proof. Recall that the complexity of a sentence is the sum of the arities of the connectives occuring within it (Definition 56). We can see by direct inspection that the tableau rules all involve a decrease in complexity. I show a couple of representative cases, letting $comp(\phi)$ stand for the complexity of ϕ:

- If $\phi = \neg\neg\psi$, then $comp(\phi) = comp(\neg\psi) + 1 = comp(\psi) + 2$, and so the complexity of the sentence licensed by the double negation rule is less than the complexity of the double negation to which it is applied.
- If $\phi = \psi \rightarrow \chi$, then $comp(\phi) = comp(\psi) + comp(\chi) + 2$. The conditional rule licenses χ on one branch, which has complexity $comp(\chi) < comp(\psi) + comp(\chi) + 2$; and licenses $\neg\psi$ on the other branch, which has complexity $comp(\psi)+1$. Even if χ is a sentence letter with complexity zero, $comp(\neg\psi) = comp(\psi) + 1 < comp(\psi) + 2 \leqslant comp(\psi) + comp(\chi) + 2 = comp(\phi)$.
- If If $\phi = \neg(\psi \wedge \chi)$, then $comp(\phi) = comp(\psi \wedge \chi) + 1 = comp(\psi) + comp(\chi) + 3$. The negated rule licenses $\neg\chi$ on one branch and $\neg\psi$ on the other. But $comp(\neg\chi) = comp(\chi)+1 < comp(\chi)+3 \leqslant comp(\psi)+comp(\chi)+3 = comp(\phi)$. Parallel reasoning shows that $comp(\neg\psi) < comp(\phi)$. □

We can now show that each finished tableau can be finite and hence a tableau – roughly because each tableau is generated by a finite generating set, in which each member has finite complexity, and the rules produce finitely many more sentences of lower complexity.

Theorem 66 (Finitude of Tableaux)
The smallest tableau generated by a finite set Γ is finite.

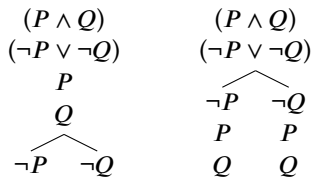
Proof. (Sketch.) Suppose we begin with a finite generating set Γ, so that every branch in the finished tableau begins with $\langle \gamma_1, \ldots, \gamma_n \rangle$. Every sentence on every branch subsequent to this initial subsequence will be the result of finitely many applications of the tableau rules to some γ_i and its further consequences. (Because the tableau is the smallest finished tableau, we do not ever apply the rules to any sentence on a branch more than once.) But each tableau rule results in extending some branch by only finitely many sentences of strictly lower complexity than the sentence to which it is applied, by Lemma 65. Since every $\mathscr{L}_1$ sentence is of finite complexity, beginning with some γ_i ends up yielding sentence letter(s) or negated sentence letter(s) after finitely many applications of the tableau rules, having produced only finitely many intermediate sentences of decreasing complexity along the way. And at that point we cannot apply the tableau rules any further without repetition, since they do not permit us to extend a branch by applying a rule to a literal. After we have exhaustively applied every tableau rule possible to each sentences in Γ and the resulting sentences, we will have produced only finitely many sentences, only finitely many times, before we run out of sentences to which we have not already applied the rules. So any finished branch beginning with Γ is finite. By König's Lemma (proved in the next chapter – Theorem 89), an infinite tableau must have an infinite branch, so any tableau with only finite branches must be itself finite. □

Accordingly, we can be sure that mechanical application of the tableau rules to a finite generating set will yield a finished finite tree which meets the conditions for being a finished tableau.

6.4 Closure and Order

Definition 97 (Closed). A branch on a finished tableau is *closed* if there is a sentence ϕ such that both ϕ and $\neg\phi$ occur on that branch; otherwise it is *open*. A tableau is closed iff every branch in it is closed.

Some people prefer to define a closed branch more strictly as one on which a *sentence letter* and its negation both appear; it is a fairly immediate though tedious consequence of the tableau rules that a finished tableau will be closed in the



(a) List then branch.

(b) Branch then list.

Figure 6.4: Two Tableaux Generated by the Same Set

stricter sense iff it is closed in our sense, because if ϕ and $\neg\phi$ occur on a branch, eventually a sentence letter from ϕ and its negation will also occur on an extension of that branch, and so will already occur on a finished branch. (The converse is trivial since sentence letters are sentences.)

Order in Tableaux We saw earlier that one and the same set can generate different tableaux, by taking the members of that set in different orders or applying the list and branch rules in different orders. An example is pictured in Figure 6.4. Does this ever matter? In particular: is there any case where one finished tableaux generated by Γ is closed, but another is not? It turns out the answer is *no*.

Compare the two tableau in Figure 6.4. They differ in which order the list and branch rules are applied. So the branches (which are just sequences coincident on their initial subsequence) differ in the order of what is on them. But the set of sentences on each branch is the same, no matter in what order the rules are applied: in both tableaux, there is a branch with the members $\{P \wedge Q, \neg(P \wedge Q), P, Q, \neg P\}$ and a branch with the members $\{P \wedge Q, \neg(P \wedge Q), P, Q, \neg Q\}$. This is entirely general.

Lemma 67 (Order and Branch Membership)
Each finished tableau generated by a finite set Γ has the same number of branches; and for each branch on any finished tableau generated by Γ, there is a branch on every other finished tableau generated by Γ with the same members though in a possibly different order.

Proof. (Sketch.) A sentence gets to be on a branch of a tableau either by being a member of the generating set Γ, or by resulting from an application of a rule to

an earlier sentence on the branch. If it is a member of the generating set, it is a member of every branch of every finished tableau generated by Γ, though it may come in a different position.

Say that Γ_{i+1} is a *tableau development* of a finite set Γ_i iff it contains Γ_i as a subset, and either (i) for some sentence in Γ_i to which a tableau list rule applies, Γ_{i+1} results from adding both listed sentences to Γ_i, or (ii) for some sentence in Γ_i to which a tableau branch rule applies, Γ_{i+1} results from adding just one of the branched sentences to Γ_i. Γ' is a *finished development* of Γ iff it is the last member of a sequence of sets beginning with Γ, each a development of its predecessor, such that any development of Γ' is Γ' itself. It is crucial to note that two sequences of developments can terminate in the same finished development.

There is a one-one correspondence between finished branches on a tableau generated by Γ and developments of Γ. This is immediate by construction. For each sentence on each branch comes from a previous 'stage' of construction of the branch by the tableau rules, and each stage corresponds to a development of the previous stage, and so each finished branch corresponds to a finished development of the initial subsequence common to all branches (the generating set). Since any tableau generated by Γ has branches in one-one correspondence with the finished developments of Γ, each such tableau has the same number of branches.

And each branch on such a tableau has a corresponding branch on any other such tableau, the one corresponding to the same development. If there were some branch B on some finished tableau $\mathbf{T}$generated by Γ that shared membership with no branch on $\mathbf{T}'$, then that branch would have as its members a finished development of Γ that was not present as a branch of $\mathbf{T}'$. Without loss of generality, there is some member of B that doesn't appear on any branch on $\mathbf{T}'$. But where did that sentence come from? Not from the generating set; and not from anything that results from an application of the rules to the generating set, since the branches are finished. But there is no other way for a sentence to get into a development, or onto the corresponding branch; so there is no such sentence. □

Since the property of being open (or closed) depends only on the members of the branch, and not on their order, rearranging the order on the branches preserves openness (and closedness). So if one tableau varies from another just in the order to which the rules have been applied to sentences on branches, then the

first is closed iff the second is closed.

Theorem 68 (Order Irrelevant)
If any finished tableau generated by Γ *closes, every finished tableau generated by* Γ *closes.*

Proof. The tableau rules are just a way of generating developments of the generating set and conveniently representing them in tree form. What matters for openness are the developments, and since we get the same developments on any finished tableau generated by Γ, by Lemma 67, then any finished tableau generated by Γ contains an open branch for any open development of Γ, and a closed branch for any closed development of Γ. So if $\mathbf{T}$ is closed, that is because every development of Γ closes, so any tableau $\mathbf{T}'$ generated by Γ is also closed. □

So really, order doesn't matter: mechanically applying the rules to Σ in any order you wish, to produce a finished tableau, will generate a closed tableau if *any* tableau generated by Σ closes. That's not to say that it doesn't matter in some sense. The tree representation of tableaux permits us to write just once any initial subsequence shared by two or more branches, and only account for the subsequent divergences by branching. If we apply list rules first where possible, then branch rules, we extend those shared initial subsequences before divergence, and thus extend the number of sentences we need to write just once. This can be readily seen in Figure 6.4, in which 6.4a contains 6 sentence tokens, while 6.4b contains 8. But the underlying sequences are two 5-membered sequences, each corresponding to the one of the two finished developments of $\{(P \wedge Q), (P \vee Q)\}$.

6.5 Tableaux Derivations

We are now in a position to describe our derivations.

Definition 98 (Tableau Derivation). A *tableau derivation* of ϕ from Γ is any finished closed tableau generated by $\Sigma \cup \{\neg\phi\}$, where Σ is a finite subset of Γ.

We say that ϕ is *derivable* from Γ, written $\Gamma \vdash \phi$, iff there is a tableau derivation of ϕ from Γ. We write $\Gamma \nvdash \phi$ to express that there is no tableau derivation of ϕ from Γ.

This is a purely formal and syntactic analogue of the notion of entailment – hence the notation which is reminiscent of the turnstile we use for entailment. Likewise there are syntactic analogues of tautologies and (un)satisifable sets.

Definition 99 (Theoremhood). We say that ϕ is a *theorem* of the tableau derivation system iff there is a finished closed tableau generated by $\{\neg\phi\}$, and we write this $\vdash \phi$.

Definition 100 (Syntactic Consistency). We say that a set of $\mathscr{L}_1$ sentences Γ is *syntactically inconsistent* iff there is a finished closed tableau generated by some finite set $\Sigma \subseteq \Gamma$, which we write $\Gamma \vdash$. A set of sentences is *syntactically consistent* iff it is not inconsistent, i.e., if every finished tableau generated by a finite subset of Γ contains at least one open branch.

These definitions are formulated to allow ϕ to be derivable from an infinite set Γ iff there is a tableau derivation of ϕ from *some* finite subset of Γ. This is as it must be, since our tableau are made of $\mathscr{L}_1$-branches, each of which is finite (Definition 89), and so the generating set must be finite. But this means that there may be more than one derivation of ϕ from Γ, each making use of different finite subsets of $\Gamma \cup \{\neg\phi\}$. Indeed, note that the definition doesn't require that we make use of all members of Γ in the generating set of a tableau derivation even when Γ is finite.

The Nature of These Definitions In a sense, these definitions don't need any further explanation. This tells you the rules for manipulating a set of $\mathscr{L}_1$ sentences in such a way as to construct a successful derivation. What *justifies* our adopting these rules? Nothing, as yet: we can treat them as *purely formal*, like the rules of chess. The rules of chess tell you what things are permissible at each stage of the game. So too, the tableau rules tell you which $\mathscr{L}_1$-trees are tableaux, given the rules governing what has to be on the branches of a tableau. When the tableau is finished, like when a game of chess is finished, one can look back and consider the aesthetics of the resulting tree or game. But there isn't much more that can be said by way of justification for why one constructed it in that way, other than that in chess one aims to finish in a winning position, and in the 'game' of tableaux,

one aims to produce a finished tableau.[2]

In another sense, of course, we want our derivations to have certain properties, and we've used some suggestive notation. We *want* every inconsistent set to be unsatisfiable; and we want every successful derivation to correspond to an entailment. So we hope that $\Gamma \vdash \phi$ iff $\Gamma \models \phi$. We don't yet know if our hopes are to be granted; perhaps we made a mistake in formulating our derivation system. In chapter 8, when we prove *soundness* and *completeness* for this derivation system, we'll see that our notation wasn't presumptuous: it is the case that whenever Γ entails ϕ, ϕ is also derivable from Γ, and *vice versa*.

Multiple Derivations Consider the generating set $\{\neg\neg(P \vee Q), \neg P, \neg Q\}$. Construction of the corresponding closed tableau is straightforward (left for exercise). Recalling Definition 98, we need to figure out how to decompose this generating set into a set of premises and a singleton set of the negated conclusion. There are three ways to do this:

1. $\{\neg\neg(P \vee Q), \neg P, \neg Q\} = \{\neg\neg(P \vee Q), \neg P\} \cup \{\neg Q\}$, so that $\neg\neg(P \vee Q), \neg P \vdash Q$;
2. $\{\neg\neg(P \vee Q), \neg P, \neg Q\} = \{\neg\neg(P \vee Q), \neg Q\} \cup \{\neg P\}$, so that $\neg\neg(P \vee Q), \neg Q \vdash P$;
3. $\{\neg\neg(P \vee Q), \neg P, \neg Q\} = \{\neg P, \neg Q\} \cup \{\neg\neg(P \vee Q)\}$, so that $\neg P, \neg Q \vdash \neg(P \vee Q)$.

All three derivations correspond to the same tableau. The tableau 'doesn't care' which are the premises and conclusions: it simply tests a set of claims for syntactic inconsistency.

6.6 Tableaux in practice

The official definition of a tableau (Definition 92) is 'static': it says that an already finished tree is a tableau iff it meets certain conditions. But the pictorial presentation of the rules suggests a dynamic route to the construction of tableaux. We begin with a finite generating set Σ. We enumerate Σ in some order – it doesn't

[2] In chess, unlike tableaux, playing one permissible move often precludes playing other permissible moves, and the game finishes before every permissible move has been played. In tableaux, there is only one permissible move at each node, and a finished tableau is such that every node which could have a rule apply to it, has had that rule applied to it. So the aesthetic interest in tableaux is restricted to the order one considers the nodes, but that isn't a huge amount of artistic freedom.

matter which, as we just saw (Theorem 68) – and then successively apply the tableau rules in some order to create a finished tableau. In this section, I will discuss some heuristics for the construction of tableaux. The metatheory of tableaux does not depend on these heuristics, nor on the various decorations on tableaux that I describe in this section. But if you are ever tasked with producing some examples of tableaux to demonstrate various claims about tableau derivability, these heuristics may be useful.

Heuristics in the Construction of Tableaux Begin by inscribing the members of your generating set, given the enumeration you have made of it – $\sigma_1, \ldots$. Inscribe σ_1 as the root node, and then each σ_i successively below it, all on the trunk with no branching.

If all the members of Σ were literals – either sentence letters or negated sentence letters – we're done: no tableau rules apply, and this is already a finished and boring tableau. But this is a rare case. Suppose then that our tableau starts out unfinished.

We will bring it closer to being finished by applying a tableaux rule to some ϕ on the tableau. This involves adding, to the bottom of *each* unfinished branch on which ϕ appears, either a sentence ψ that is a constituent of ϕ, or the negation of ψ. If the rule applied is a list rule, we will add perhaps one but typically two sentences to the bottom of each branch; if the rule applied is a branch rule, we will create new branches by adding two nodes at the bottom of each branch on which ϕ appears and inscribing the appropriate sentences, one in each node.

Once we've applied a rule to a sentence, and added the relevant sentences, we're done with that sentence. We can informally indicate that we're done with a sentence by drawing a box around it.[3] But these boxes – as well as the signs to indicate a closed branch – are useful scaffolding for the construction of a tableau by a person, not part of the official definition. We can then return to apply the tableau rules to a sentence at a node not previously dealt with.

We know already that this process finishes after a finite time, even carried out completely mechanically. But we ought not be completely mechanical. We should apply list rules before branch rules to save unnecessarily early divergence

[3]This is the convention adopted by Beall and van Fraassen (2003). Others use check marks to indicate that a sentence at a node has been 'dealt with' (Jeffrey, 2006).

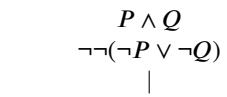
of branches. If you can see that a sentence on your branch has a constituent that would cause a branch on which it appears to close, you may as well apply the rule to it and close the branch. The branch may not be finished, but since an unfinished closed branch stays closed even if finished, you do not need to write out all the further nodes of a closed branch – if what you care about is tableau derivations, rather than the construction of finished tableaux for their own sake.

A Worked Example I illustrate this procedure, for an example generating set, in Figure 6.5. Begin by writing down the generating set $\{P \wedge Q, \neg\neg(\neg P \vee \neg Q)\}$, and take the first sentence of that tableau under this enumeration, $P \wedge Q$. It is a conjunction; by the rules for conjunctions, we know that if this sentence occurs on the tableau, both conjuncts can be inscribed, extending the branch, as in Figure 6.5(b). Next we turn to $\neg\neg(\neg P \vee \neg Q)$; we see by the rules for negation, a double negation permits the inscription of the enclosed sentence without the two negation signs. So we inscribe $\neg P \vee \neg Q$, and box the original sentence, as in Figure 6.5(c).

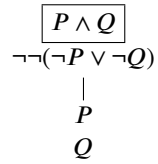
Now we come to something different. The rules for disjunction are branching rules: they don't tell us to inscribe new sentences on the same branch, but to inscribe one sentence on one branch, and another sentence on another: one for each disjunct, as in Figure 6.5(d).

Having applied the branching rule and created two branches on this tableau, we can break the sentences in the tableau down no further: we have the smallest (possibly negated) constituents of the sentences in question. So this tableau is finished – anything else licensed by the tableau rules would simply repeat something occuring in an earlier node. We see, now, if the tableau is closed. And we can see that every branch – which runs all the way from the root to the leaf node – is in fact closed. We mark this, unofficially, by inscribing $\otimes$ beneath the terminal leaf node of each closed branch in Figure 6.6(a). (Again, this is decoration; really, a branch is closed iff a sentence and its negation both occur, whether we write the $\otimes$ or not.) Since the tableau is finished and closed, this is a tableau derivation of $\neg(\neg P \vee \neg Q)$ from $(P \wedge Q)$ – a tableau derivation of one of the De Morgan equivalences (Theorem 52).

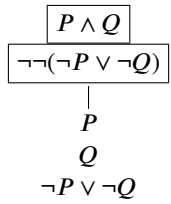
Compare the result to the tableau from Figure 6.6(b); here, the right-most branch is open, and no unboxed sentences occur earlier on the branch that would license anything new to potentially close it. So this tableau is finished and open; it



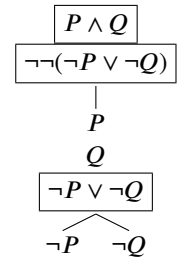
(a) Setting out the set to generate a tableau.



(b) First step: conjunction.



(c) Dealing with double negation.



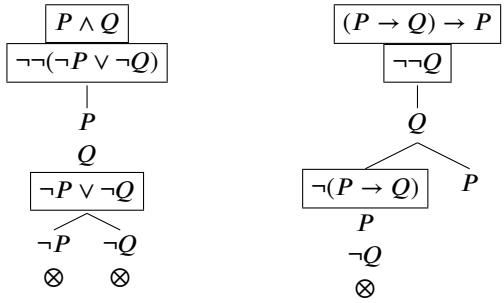
(d) Dealing with disjunction.

Figure 6.5: Steps in the construction of a tableau.

is not a derivation of $\neg Q$ from $(P \to Q) \to P$, and because any tableau generated by that set will have the same properties as this one, there is no such derivation: $(P \to Q) \to P \nvdash \neg Q$.[4]

Schematic Tableaux Consider the schematic tableau (schematic, because it involves variables over sentences ϕ, ψ, etc., rather than particular sentences P, Q, etc.) pictured in Figure 6.7. This demonstrates that $\phi \to \psi, \psi \to \chi \vdash \phi \to \chi$, for any ϕ, ψ, χ. So any tableau derivation involving sentences of this form – any uniform substitution of $\mathscr{L}_1$ sentences for these variables into this tableau – will also be closed, and the corresponding derivation exists. Note that I don't bother to add new sentences below the occurrence of χ on the rightmost branch even though strictly speaking our mechanical tableau construction procedure requires it: that branch is closed, so every extension of it would remain closed, as adding

[4]In fact, we can read off from the open branches a structure - the one where all sentence letters occuring on the branch are assigned T and all negated sentence letters are assigned F – where the premise is true and the conclusion false. So when P and Q are both true, the conclusion is false, but the premise true. This fact will be of significance in the next chapter when we prove completeness.



(a) Closing the tableau from Figure 6.5.

(b) An open tableau.

Figure 6.6: Open and closed tableaux.

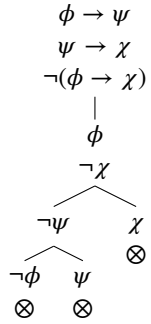
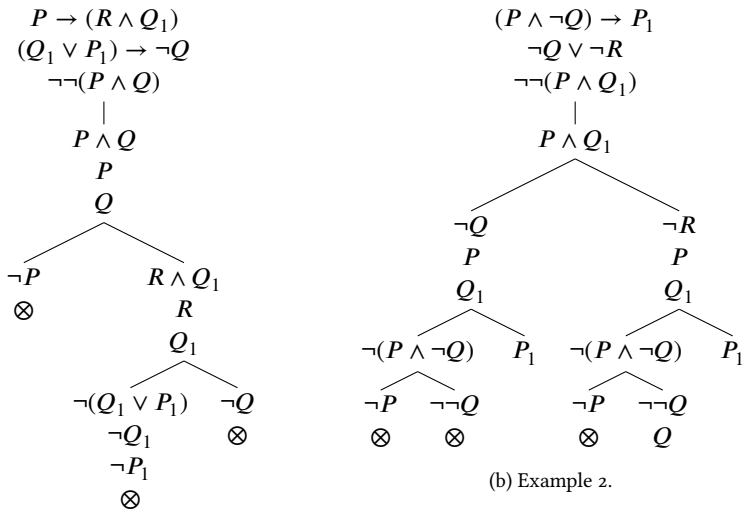


Figure 6.7: Tableaux for $\phi \to \psi, \psi \to \chi \vdash \phi \to \chi$.

new sentences to an inconsistent set cannot render it consistent. This is even more sensible if the sentence ϕ is itself subject to tableau rules – the schematic tableau shows we never need to apply any rules to ϕ, even if some apply, to determine whether the corresponding derivation exists.

Two Further Examples

1. Consider the generating set $\{P \to (R \wedge Q_1), (Q_1 \vee P_1) \to \neg Q, \neg\neg(P \wedge Q)$. The tableau is shown in Figure 6.8(a). This tableau is closed; hence there is a derivation corresponding to our original generating set: $P \to (R \wedge Q_1), (Q_1 \vee P_1) \to \neg Q, \vdash \neg(P \wedge Q)$.



(a) Example 1.

(b) Example 2.

Figure 6.8: Tableaux for our further examples.

2. Consider the argument $(P \wedge \neg Q) \to P_1; \neg Q \vee \neg R$; therefore $\neg(P \wedge Q_1)$. The tableau is pictured in Figure 6.8(b). As is readily seen, this tableau is open; there is no successful derivation corresponding to this argument.

Further Reading

Tableaux were introduced in roughly the form discussed here by Smullyan (1968). Other texts which use various forms of tableaux for sentential logic are Beall and van Fraassen 2003; Bostock 1997; Hodges 2001; Jeffrey 2006; Priest 2008 and Smith 2012.

Exercises

1. Construct a tableau for the generating set $\{\neg\neg(P \vee Q), \neg P, \neg Q\}$, mentioned on page 123.
2. Provide tableaux derivations demonstrating the following:
   (a) $\vdash ((P \leftrightarrow (P \vee P)) \wedge (P \leftrightarrow (P \wedge P)))$;
   (b) $\vdash ((P \leftrightarrow Q) \leftrightarrow ((P \to Q) \wedge (\neg Q \vee P)))$;

(c) $\vdash ((P \to Q) \leftrightarrow ((P \wedge \neg Q) \to (P \wedge \neg P)))$;

(d) $\vdash (((P \to Q) \wedge \neg Q) \to \neg P)$;

(e) $\vdash ((P \to Q) \to ((P \vee R) \to (Q \vee R)))$.

3. Provide tableaux derivations demonstrating the following:

(a) $P \vee (Q \wedge R) \vdash (P \vee Q) \wedge (P \vee R)$;

(b) $(P \wedge Q) \vee (P \wedge R) \vdash P \wedge (Q \vee R)$;

(c) $P \vdash (Q \leftrightarrow \neg Q) \to \neg P$;

(d) $(P \to Q) \to Q \vdash P \vee Q$;

(e) $P \leftrightarrow Q \vdash \neg(P \wedge Q) \to (\neg P \wedge \neg Q)$.

4. Construct finished open tableaux demonstrating the following;

(a) $((P \wedge R) \to Q) \nvdash R \wedge (P \to Q)$;

(b) $(P \vee Q), P \nvdash \neg Q$;

(c) $(P \leftrightarrow Q), (R \to P) \leftrightarrow (Q \to R)$.

5. Show the remaining cases in the proof of Lemma 65.

6. Prove that if $\vdash \phi$, then for any ψ, $\vdash \psi \to \phi$.

Answers to selected exercises on page 248.

Chapter 7

Tableau Metatheory: Soundness and Completeness

7.1 Derivations and Semantic Arguments

At the beginning of the previous chapter, we mentioned a number of considerations that motivate the development of formal object language derivation systems. Now that we have an example of such a system for $\mathscr{L}_1$, we may reflect on how it compares to evaluating arguments by directly discussing $\mathscr{L}_1$-structures.

One thing to note is that the tableau system can be considerably more efficient. Consider this argument:

$$(P \wedge (Q_1 \wedge (Q_2 \wedge (Q_3 \wedge (Q_4 \wedge Q_5))))) \vdash (P \vee (R_1 \vee (R_2 \vee (R_3 \vee (R_4 \vee R_5))))).$$

We can see the tableau derivation demonstrating the correctness of this claim in Figure 7.1 closes vey quickly. Even if we finish the tableau by applying the conjunction and negated disjunction rules repeatedly, the whole thing will be fewer than 30 lines.

It is easy to see – by giving a semantic argument that basically parallels this tableau derivation – that this argument is a valid entailment. But demonstrating this via truth tables would involve a truth table of $2^{11} = 2048$ rows, which is

$$
\begin{array}{c}
P \wedge (Q_1 \wedge (Q_2 \wedge (Q_3 \wedge (Q_4 \wedge Q_5)))) \\
\neg(P \vee (R_1 \vee (R_2 \vee (R_3 \vee (R_4 \vee R_5))))) \\
| \\
P \\
(Q_1 \wedge (Q_2 \wedge (Q_3 \wedge (Q_4 \wedge Q_5)))) \\
\neg P \\
\neg(R_1 \vee (R_2 \vee (R_3 \vee (R_4 \vee R_5)))) \\
\otimes
\end{array}
$$

Figure 7.1: A brief tableau.

prohibitively unwieldy. The tableau procedure is no less mechanical than the truth table procedure, but is considerably more efficient in this case.

On the other hand, consider an invalid argument like

$$(P \vee (\neg P \vee (\neg\neg P \vee (\neg\neg\neg P \vee \neg\neg\neg\neg P)))) \not\models (P \wedge (\neg P \wedge (\neg\neg P \wedge (\neg\neg\neg P \wedge \neg\neg\neg\neg P)))).$$

The truth table is two lines, on each of which the premise is true and the conclusion false. But the tableau, with its many uses of the disjunction and negated conjunction rules, branches extensively and is much less convenient to work with.

This pattern is fairly representative. To show an argument invalid, we need to demonstrate the existence of just one structure in which the premises are true and the conclusion false. But checking invalidity by the construction of an attempted derivation involves the construction of a finished open tableau. In bad cases, this construction will take a long time to complete. To show an argument valid involves demonstrating the non-existence of a counterexample structure, which may require consideration of a great many structures. But a direct derivation of the conclusion from the premises may be more efficient.

We cannot yet recommend that we use semantic techniques to demonstrate invalidity, and formal derivations to demonstrate validity. This is because we do not yet know whether *every* valid argument has a corresponding derivation; nor do we know whether *every* derivation corresponds to a valid argument. It turns out that every valid argument is derivable, and every correctly constructed derivation can be associated with a valid argument. Both of these facts will be proved later in this chapter (section 7.3), when we discuss the soundness and completeness of the tableau derivation system with respect to the semantics of $\mathscr{L}_1$.

7.2 Transforming Derivations

Sometimes when we have a tableau derivation, we can use purely formal manipulations on the tree to construct other derivations. While trees can and do have other trees occuring within them (Def. 91), the trees which occur within a given tableau will always not meet the conditions for being a tableau themselves. But often simple modifications will enable us to convert one tableau into another, taking sub-trees of a tableau and re-using them in a new tableau.

Structural Rules We proved some structural rules for $\vDash$ (section 4.8). We can show that analogous rules apply to $\vdash$. (I leave the demonstration of weakening for an exercise.)

Theorem 69 ($\vdash$ Permutation)
$\Gamma, \psi, \chi, \Delta \vdash \phi$ *iff* $\Gamma, \chi, \psi, \Delta \vdash \phi$.

Proof. Immediate consequence of Theorem 68. □

Theorem 70 ($\vdash$ Contraction)
$\Gamma, \psi, \psi \vdash \phi$ *iff* $\Gamma, \psi \vdash \phi$.

Proof. $\Gamma, \psi, \psi \vdash \phi$ iff there is a finished closed tableau generated by some finite set $\Sigma \cup \{\psi, \psi, \neg\phi\}$, where $\Sigma \subseteq \Gamma$. But any tableau generated by $\Sigma \cup \{\psi, \psi, \neg\phi\}$ is generated by $\Sigma \cup \{\psi, \neg\phi\}$, and vice versa, because those generating sets are identical (Def. 93). □

Syntactic Deduction Just as we proved the deduction theorem (Theorem 33) for the semantic turnstile, we can prove a syntactic deduction theorem for the syntactic turnstile.

Theorem 71 (Syntactic Deduction)
$\Gamma, \phi \vdash \psi$ *iff* $\Gamma \vdash \phi \to \psi$.

Proof. If: Assume $\Gamma \vdash \phi \to \psi$. Then there is at least one finite set Σ, a subset of Γ, such that every finished tableau $\mathbf{T}$ which is generated by $\Sigma \cup \{\neg(\phi \to \psi)\}$ is closed. Take one such tableau, where the first tableau rule applied was the negated conditional rule (Figure 6.2(f)), so that each branch of the tableau begins $\langle \gamma_1, \ldots, \gamma_n, \neg(\phi \to \psi), \phi, \neg\psi \rangle$, where each $\gamma_i \in \Sigma \subseteq \Gamma$. We have already dealt

with $\neg(\phi \rightarrow \psi)$, so the sentences which close every branch on this tableau derive from Σ or from ϕ or $\neg\psi$. Hence, we can construct a closed tableau where every branch begins $\langle \gamma_1, \ldots, \gamma_n, \phi, \neg\psi \rangle$; hence some finite subset of $\Gamma \cup \{\phi, \neg\psi\}$ generates a closed tableau, hence $\Gamma, \phi \vdash \psi$. The only interesting case is if the original tableau had a branch which closed because of the presence of $(\phi \rightarrow \psi)$ on it. But then an application of the branching conditional rule to that node gives one branch with $\neg\phi$ occurring on it, which closes since ϕ occurs in the trunk; and another branch with ψ occurring on it, which closes since $\neg\psi$ occurs in the trunk.
Only if: We assume that $\Gamma, \phi \vdash \psi$. Then there is a closed finished tableaux that is generated by a finite subset of $\Gamma \cup \{\phi, \neg\psi\}$. If we modify this tableau by inserting $\neg(\phi \rightarrow \psi)$ above ϕ on every branch, the modified tree is closed also. Since ϕ and $\neg\psi$ can come from $\neg(\phi \rightarrow \psi)$ (Figure 6.2(f)), this modified tree in fact satisfies the conditions for being a tableau. Since the original tableau is finished, and the modified tableau contains the results of applying tableau rules to the only new node, this is a finished tableau generated by a finite subset of $\{\Gamma, \neg(\phi \rightarrow \psi)\}$, which means that $\Gamma \vdash \phi \rightarrow \psi$. □

Now I'll prove some lemmas preliminary to proving a historically important theorem, Cut (Theorem 76). All of the following four lemmas involve showing that some tableau derivations can be converted into other tableau derivations, in such a way that the syntactic derivability mirrors what is intuitively provable in virtue of the meaning of the sentences involved.

Lemma 72
$\Gamma \vdash \phi$ *iff* $\Gamma, \neg\phi \vdash$.

Proof. $\Gamma \vdash \phi$ iff there exists a finite set $\Sigma \subseteq \Gamma$ such that there is a finished closed tableau generated by $\Sigma \cup \{\neg\phi\}$. This holds in turn iff $\Sigma \cup \{\neg\phi\}$ is inconsistent (by Definition 100). □

Lemma 73
$\Gamma, \neg\neg\phi \vdash$ *iff* $\Gamma, \phi \vdash$.

Proof. Only If: If $\Gamma, \neg\neg\phi \vdash$, there is a finished closed tableau generated by $\Sigma \cup \{\neg\neg\phi\}$, where Σ is a finite subset of Γ. Take such a tableau in which the first rule applied to the generating set was the $\neg\neg$ rule (Figure 6.2(i)). Remove the node containing $\neg\neg\phi$: the tableau will still be closed. Why? Either Σ itself already contained

some sentence and its negation; or the tableau closed because of some application of a tableau rule to ϕ (the result of applying the double negation rule to $\neg\neg\phi$); or the tableau closed because $\neg\neg\phi$ and its negation occurred on the tableau. In the first two cases, it is obvious that removing $\neg\neg\phi$ made no difference to whether the tableau is closed. So consider the third case. There are two subcases: either $\neg\neg\neg\phi$ occurs on the tableau, or $\neg\phi$ does (both are negations of $\neg\neg\phi$). If the latter, the tableau is closed because both ϕ and $\neg\phi$ occur. If the former, since the tableau is finished, some application of the double negation rule was made to $\neg\neg\neg\phi$ at some stage, so that $\neg\phi$ occurs as well as ϕ, again closing the relevant branch.
If: If $\Gamma, \phi \vdash$, there is a finished closed tableau generated by $\Sigma \cup \{\phi\}$. Modify this tableau by inserting $\neg\neg\phi$ above the node containing ϕ. The resulting tree remains closed; it also satisfies the conditions for being a tableau generated by $\Sigma \cup \{\neg\neg\phi\}$, since ϕ comes from an earlier node by application of the double negation rule. So $\Gamma, \neg\neg\phi \vdash$. □

Lemma 74
If $\Gamma \vdash \neg(\phi \wedge \psi)$ *then* $\Gamma, \phi \vdash \neg\psi$.

Proof. If $\Gamma \vdash \neg(\phi \wedge \psi)$, then there is a finite $\Sigma \subseteq \Gamma$ such that every finished tableau generated by $\Sigma \cup \{\neg\neg(\phi \wedge \psi)\}$ is closed, and by Lemma 73, so too is every finished tableau generated by $\Sigma \cup \{\phi \wedge \psi\}$. Consider such a tableau in which the first rule applied was the conjunction rule (Figure 6.2(a)); this is also a finished closed tableau generated by $\Sigma \cup \{\phi \wedge \psi, \phi, \psi\}$. Remove the node containing $\phi \wedge \psi$, and the resulting tree remains a closed tableau. Either the tableau is closed because of the results of applying the tableau rules to ϕ and ψ; or some branch of the tableau is closed because it contains $\neg(\phi \wedge \psi)$. But, because the tableau is finished, any such branch also contains either $\neg\phi$ or $\neg\psi$; either way, it will then close because of the presence of ϕ and ψ earlier. So $\Gamma, \phi, \psi \vdash$; by Lemma 73, $\Gamma, \phi, \neg\neg\psi \vdash$; by Lemma 72, $\Gamma, \phi \vdash \neg\psi$. □

Lemma 75
If $\Gamma \vdash (\phi \wedge \psi)$, *then* $\Gamma \vdash \phi$ *and* $\Gamma \vdash \psi$.

Proof. If $\Gamma \vdash (\phi \wedge \psi)$, there is a finished closed tableau generated by $\Sigma \cup \{\neg(\phi \wedge \psi)\}$ where Σ is a finite subset of Γ. Suppose the first rule applied was the negated conjunction rule (Figure 6.2(b)), giving two branches, both beginning with the

members of Σ and $\neg(\phi \wedge \psi)$, but one continuing with $\neg\phi$, and the other with $\neg\psi$. Because the tableau is closed, every branch beginning like those two branches is closed. Suppose we delete the node containing $\neg(\phi \wedge \psi)$, but keep $\neg\phi$. We will also therefore prune off the branch with $\neg\psi$ occurring on it. But the pruned tree remains a closed tableau, either because the tableau branches close due to the application of some tableau rule to $\neg\phi$, or because $\phi \wedge \psi$ occurs somewhere on the tableau; since it is finished, ϕ also appears, which closes the branch. The same goes, *mutatis mutandis*, for the other pruning. So $\Gamma \vdash \phi$ and $\Gamma \vdash \psi$. □

Now we are in a position to prove the Cut theorem (or at least one theorem that commonly goes under that name.)

Theorem 76 (Cut)
If $\Gamma \vdash \phi$ *and* $\Gamma \vdash \neg\phi$ *then* $\Gamma \vdash$.

Proof. We prove by (strong) induction on the complexity of ϕ.
Base case: suppose ϕ is a sentence letter. Then there is a finite $\Pi_1 \subseteq \Gamma$ such that $\Pi_1 \cup \{\neg\phi\}$ generates a finished closed tableau, and another finite $\Pi_2 \subseteq \Gamma$ such that $\Pi_2 \cup \{\neg\neg\phi\}$ generates a finished closed tableau. Let $\Sigma = \Pi_1 \cup \Pi_2$ – Σ too will be a finite subset of Γ. And note that adding more sentences to the trunk of a finished closed tableau will render it unfinished, but it will remain closed – finishing it will mean adding new leaves and possibly branches to already closed branches. So we can conclude that there is a finished closed tableau generated by $\Sigma \cup \{\neg\phi\}$, and a finished closed tableau generated by $\Sigma \cup \{\neg\neg\phi\}$.

Suppose there is a finished open tableau $\mathbf{T}$ generated by Σ alone. Inserting ϕ at the base of the trunk of $\mathbf{T}$ must render it closed, which means that each open branch must already have contained $\neg\phi$. But since each tableau generated by $\Sigma \cup \{\neg\phi\}$ is also closed, those branches must already have contained ϕ – those branches cannot in fact have been open. So any tableaux generated by Σ must be closed, and hence Σ is inconsistent and so are all of its supersets, including Γ.
The induction step: ϕ is complex, and the induction hypothesis is that for each less complex constituent ψ of ϕ, and for any set Δ such that $\Delta \vdash \psi$ and $\Delta \vdash \neg\psi$, then $\Delta \vdash$. I consider two cases explicitly, those where ϕ is either a negation or a conjunction:

1. $\phi = \neg\psi$. We assume $\Gamma \vdash \phi$ and $\Gamma \vdash \neg\phi$, i.e., by Lemma 72, $\Gamma, \phi \vdash$ and $\Gamma, \neg\phi \vdash$. Because $\phi = \neg\psi$, we also have $\Gamma, \neg\psi \vdash$ and $\Gamma, \neg\neg\psi \vdash$. By Lemma

73, $\Gamma, \psi \vdash$. By the induction hypothesis applied to the simpler constituent ψ, $\Gamma \vdash$.

2. $\phi = (\psi \wedge \chi)$. Again, we assume (a) $\Gamma \vdash (\psi \wedge \chi)$ and (b) $\Gamma \vdash \neg(\psi \wedge \chi)$. By Lemma 75 applied to (a), we have $\Gamma \vdash \chi$. But if a tableau generated by $\Sigma \cup \{\neg\chi\}$ closes, so does one generated by $\Sigma \cup \{\psi, \neg\chi\}$, where Σ is a finite subset of Γ. So $\Gamma, \psi \vdash \chi$. By Lemma 74 applied to (b), we have $\Gamma, \psi \vdash \neg\chi$. By the induction hypothesis $\Gamma, \psi \vdash$. But from Lemma 75 applied to (a), we also have $\Gamma \vdash \psi$; by Lemma 72, $\Gamma, \neg\psi \vdash$; and by the induction hypothesis, $\Gamma \vdash$.

3. Other cases left for exercises; you will also need to establish lemmas parallelling Lemmas 74–75 for disjunction, conditional, and biconditional. □

What does the Cut Theorem show? Basically this: if you can get a tableau generated by Γ and $\neg\phi$ to close, and you can get a tableau generated by Γ and ϕ to close, then you could already have got a tableau generated by Γ to close: whatever appeal you made respectively to ϕ and $\neg\phi$, it was inessential that they appeared on the trunk. (Either they weren't used, or they were derivable from something else already in Γ.) The interest of Cut in derivation systems is in showing that our derivations can all be 'direct' in a sense: if we can derive something with the assistance of either ϕ or $\neg\phi$, then we can derive it without detouring through that assistance.[1] Here is another way of making the same point.

Theorem 77 (Cut Again)
If $\Gamma, \phi \vdash \psi$ *and* $\Gamma \vdash \phi$ *then* $\Gamma \vdash \psi$.

Proof. Suppose $\Gamma, \phi \vdash \psi$; then $\Gamma \cup \{\neg\psi\}, \phi \vdash$ by Lemma 72 (and the fact that what flanks the turnstile are sets in which order doesn't matter). Suppose $\Gamma \vdash \phi$; then $\Gamma, \neg\phi \vdash$, and $\Gamma \cup \{\neg\psi\}, \neg\phi \vdash$ (by Weakening for $\vdash$, proved in exercises). By Theorem 76, $\Gamma \cup \{\neg\psi\} \vdash$, i.e., $\Gamma \vdash \psi$. □

In fact, this theorem is equivalent to Theorem 76. Note the parallel with Theorem 38.

[1] This is more important when we consider natural deduction derivations in Chapters 8–9, where the Cut theorem is used to show that in principle all derivations can have a particularly elegant form where sentences are first broken down into simpler constituents and then built back up to establish the desired conclusions.

Theorem 78 (Transitivity of Derivability)
If $\Gamma \vdash \phi$ *and* $\phi \vdash \psi$, *then* $\Gamma \vdash \psi$.

Proof. Assume $\phi \vdash \psi$. Then by weakening $\Gamma, \phi \vdash \psi$. Assume $\Gamma \vdash \phi$. Then by Theorem 77, $\Gamma \vdash \psi$. □

This shows is that any two-step derivation has a short-cut: if we can derive ψ from something we can derive from Γ, well, we could have derived ψ directly.

7.3 Soundness and Completeness

Soundness and completeness are properties that a specified derivation system for sentences in a given language may or may not have with respect to a particular semantic interpretation of the language.

Definition 101 (Soundness). A derivation system P in a given language is *sound* with respect to a semantic interpretation of the language just in case whenever there is a derivation which establishes $\Gamma \vdash_P \phi$, it is the case that $\Gamma \vDash \phi$.

Definition 102 (Completeness). A derivation system P in a given language is *complete* with respect to a semantic interpretation of the language just in case whenever it is the case that $\Gamma \vDash \phi$, there is a derivation which establishes $\Gamma \vdash_P \phi$.

Obviously we've already talked informally about the correspondence between $\vDash$ and $\vdash$ we are about to establish, and we designed our derivation system with one eye on capturing all and only correct entailments in derivations. But those intentions in the design of these derivation systems could have gone awry, so it is important to check they have not. So we prove soundness for our tableau derivation system.

7.4 Soundness of the Tableaux Derivation System

We begin by proving soundness of the tableaux system. We will prove it in this form: that every syntactic theorem – derivable sentence – is a semantic theorem, so that if $\vdash \phi$ then $\vDash \phi$.[2]

[2]Other proofs of soundness for tableau derivation systems can be found in Bostock (1997: 165–7), Hodges (2001: 119), and Jeffrey (2006: 33–4), among many others.

Lemma 79 (Tableau rules preserve satisfiability)
If a tableau is generated by a satisfiable negated sentence, there is at least one branch on that tableau such that every sentence on that branch is simultaneously satisfiable. That is: if there is an $\mathscr{L}_1$ structure $\mathscr{A}$ such that $[\![\neg\phi]\!]_{\mathscr{A}} = T$, there is some branch B on a tableau generated by $\{\neg\phi\}$ such that for all sentences $\beta \in B$, $[\![\beta]\!]_{\mathscr{A}} = T$.

Proof. We prove the lemma by *induction on the length of tableau branch B*. In effect, we consider a sequence of (mostly unfinished) tableaux, such that each member extends the preceding member by application of one of the tableau rules.

Base case: Consider the smallest tableaux $\mathbf{T}$ generated by $\{\neg\phi\}$: the single node $\langle\neg\phi\rangle$. Since this is the only member of the only branch on $\mathbf{T}$, and by hypothesis it is assigned T by $\mathscr{A}$, the lemma holds in this case.

Induction step: Assume that the lemma holds of a branch B on tableau $\mathbf{T}_n$. Then we show the lemma holds of a branch B^+ on a tableau $\mathbf{T}_{n+1}$ obtained from $\mathbf{T}_n$ by one additional application of a rule in Figure 6.2 to a sentence in B on T_n. There are three cases.

1. We apply the Double Negation Rule (Figure 6.2(i)). Then some sentence on B is of the form $\neg\neg\chi$, and we add χ to the bottom of the branch to get B^+. Since the valuation function induced by the $\mathscr{L}_1$ structure $\mathscr{A}$ is classical, $[\![\chi]\!]_{\mathscr{A}} = [\![\neg\neg\chi]\!]_{\mathscr{A}} = T$, as required by the lemma.

2. We apply a non-branching rule to some sentence on B. Then we add two sentences to the bottom of B. If we applied, for example, the Negated Conditional rule (Figure 6.2(f)) to $\neg(\psi \to \chi)$, we added ψ and $\neg\chi$ to the bottom of B to get B^+. By the rules on classical valuations, $[\![\neg(\psi \to \chi)]\!]_{\mathscr{A}} = T$ iff $[\![\psi \to \chi]\!]_{\mathscr{A}} = F$ iff $[\![\psi]\!]_{\mathscr{A}} = T$ and $[\![\chi]\!]_{\mathscr{A}} = F$ iff $[\![\psi]\!]_{\mathscr{A}} = T$ and $[\![\neg\chi]\!]_{\mathscr{A}} = T$, as required.

   Similarly for the Conjunction rule: $[\![\psi \wedge \chi]\!]_{\mathscr{A}} = T$ iff $[\![\psi]\!]_{\mathscr{A}} = T$ and $[\![\chi]\!]_{\mathscr{A}} = T$. So the lemma holds of B^+. (The Negated Disjunction rule is left as an exercise.)

3. We apply a branching rule to some sentence on B. We then get two new branches, B_1^+ and B_2^+, either of which can satisfy the lemma (but we only need one of them to satisfy it: the lemma only says that there is *at least one* branch on a tableau such that all of the sentences on it are assigned T by $\mathscr{A}$).

For instance, if we apply Disjunction (Figure 6.2(c)) to $\psi \vee \chi$, $B_1^+ = B \cup \{\psi\}$ and $B_2^+ = B \cup \{\chi\}$. Since by the rules for classical valuations, $[\![\psi \vee \chi]\!]_{\mathscr{A}} = T$ iff either $[\![\psi]\!]_{\mathscr{A}} = T$ or $[\![\chi]\!]_{\mathscr{A}} = T$; that is, the lemma holds of at least one of B_1^+ or B_2^+.

Similarly for the Negated Conjunction rule: If $\neg(\phi \wedge \psi) \in B$, then $\neg\phi \in B_1^+$ and $\neg\psi \in B_2^+$. $[\![\neg(\phi \wedge \psi)]\!]_{\mathscr{A}} = T$ iff $[\![\phi \wedge \psi]\!]_{\mathscr{A}} = F$ iff $[\![\phi]\!]_{\mathscr{A}} = F$ or $[\![\psi]\!]_{\mathscr{A}} = F$ iff $[\![\neg\phi]\!]_{\mathscr{A}} = T$ or $[\![\neg\psi]\!]_{\mathscr{A}} = T$. (The Conditional rule is left as an exercise.)

That completes the induction. □

Lemma 79 shows that the tableau rules preserve satisfiability: any tableau, finished or otherwise, generated by a satisfiable sentence has at least one branch, all of whose elements are simultaneously satisfiable. Now we are in a position to prove soundness.

Theorem 80 (Soundness)
If $\vdash \phi$ *then* $\vDash \phi$.

Proof. Assume that $\vdash \phi$. Then every finished tableau generated by $\{\neg\phi\}$ is closed.

Assume for *reductio* that $\nvDash \phi$. Then there is an $\mathscr{L}_1$ structure $\mathscr{A}$ that makes $\neg\phi$ true: $[\![\neg\phi]\!]_{\mathscr{A}} = T$. By Lemma 79, there is a finished tableau $\mathbf{T}$ generated by $\neg\phi$, with a branch B such that for every $\beta \in B$, $[\![\beta]\!]_{\mathscr{A}} = T$. Since the valuation function induced by $\mathscr{A}$ is classical, for any $\mathscr{L}_1$ sentence ψ, if $\psi \in B$, then $\neg\psi \notin B$. (Since for any structure A, $[\![\psi]\!]_{\mathscr{A}} \neq [\![\neg\psi]\!]_{\mathscr{A}}$, and every sentence on B is satisfied in some structure and hence all have the same truth value in that structure.) Hence B cannot be closed; hence $\mathbf{T}$ is not closed. But $\mathbf{T}$ is generated by $\{\neg\phi\}$, and our initial assumption was that any finished tableau generated by $\{\neg\phi\}$ is closed. So our *reductio* hypothesis must be wrong; that is, it must instead be true that $\vDash \phi$. But now we've shown $\vDash \phi$ on the assumption that $\vdash \phi$. □

Having proved the theorem, we can easily extend it to the general case of an arbitrary argument.

Theorem 81 (General Soundness)
If $\Gamma \vdash \phi$ *then* $\Gamma \vDash \phi$.

Proof. Recall that $\Gamma \vdash \phi$ iff there is a finished closed tableau generated by a finite set $\Sigma \cup \{\neg\phi\}$, where $\Sigma \subseteq \Gamma$, i.e., iff $\Sigma \vdash \phi$.

Assume that $\Gamma \vdash \phi$. Then there is a finite set $\Sigma \cup \{\neg\phi\} = \{\sigma_1, \ldots, \sigma_n, \neg\phi\}$ which generates a finished closed tableau. By repeated applications of the Deduction theorem for tableaux (Theorem 71), $\vdash (\sigma_1 \to (\sigma_2 \to (\ldots(\sigma_n \to \phi))))$. By Soundness (Theorem 80), $\vDash (\sigma_1 \to (\sigma_2 \to (\ldots(\sigma_n \to \phi))))$. By repeated applications of the Deduction theorem for entailment (Theorem 33), $\Sigma \vDash \phi$. Since $\Sigma \subseteq \Gamma$, $\Gamma \vDash \phi$, by Weakening for $\vDash$.[3] □

7.5 Completeness for Tableaux

The proof of completeness for the tableaux derivation system is relatively straightforward. The idea is simple: we show that, if set of sentences Γ is consistent, then Γ is satisfiable. Recall that (Definition 100) a set of sentences Γ is syntaxctically consistent if every finished tableau generated by a finite subset of Γ has an open branch. We will use this fact to show that there is a structure in which all the members of Γ are true. To do this, we have to show an intermediate lemma, to the effect that an open branch can be used to determine a structure which makes each sentence on the branch true.

Lemma 82 (Hintikka's Lemma)
If B is an open branch on a finished open tableau generated by a finite set of $\mathscr{L}_1$ sentences Σ, then there is an $\mathscr{L}_1$ structure $\mathscr{B}$ such that for every sentence β on B, $[\![\beta]\!]_{\mathscr{B}} = T$.

Proof. Suppose there is an open branch on a finished tableau generated by Σ, B. Define an $\mathscr{L}_1$ structure $\mathscr{B}$ as follows: for every sentence letter s,

$$\mathscr{B}(s) = \begin{cases} T & \text{if } s \text{ occurs in some node on } B; \\ F & \text{if } \neg s \text{ occurs in some node on } B; \\ T & \text{otherwise.} \end{cases}$$

[3] Actually (because Weakening as stated only applies to a single sentence) what we need is the stronger claim (still obviously correct), that for any set Δ, if $\Sigma \vDash \phi$, then $\Sigma \cup \Delta \vDash \phi$. And we then let $\Delta = \Gamma \setminus \Sigma$, so that $\Sigma \cup \Delta = \Gamma$, and the result follows.

Because $\boldsymbol{B}$ is an open branch, this successfully defines an $\mathscr{L}_1$ structure: no sentence letter and its negation both occur on $\boldsymbol{B}$, so this definition assigns one and only one value to each $\mathscr{L}_1$ sentence letter. (It is thus a classical structure, and induces a classical two-valued valuation of all sentences.)

We show by induction on complexity on sentences that for every sentence β on $\boldsymbol{B}$, $[\![\beta]\!]_{\mathscr{B}} = \boldsymbol{T}$. Because this is a tableau construction, the neatest way to think about the induction on complexity is that the base case comprises those sentences to which no tableau rule applies, and the induction step involves consideration of sentences to which tableau rules apply. (Our base 'case' thus encompasses literals, rather than just sentence letters.)

Base case: Suppose β occurs on $\boldsymbol{B}$, and is a literal: either is a sentence letter or a negated sentence letter. If the former, by construction, $\mathscr{B}(\beta) = \boldsymbol{T}$, so $[\![\beta]\!]_{\mathscr{B}} = \boldsymbol{T}$, by the definition of an $\mathscr{L}_1$ structure. If the latter, i.e., $\beta = \neg s$ for some sentence letter s, $[\![\beta]\!]_{\mathscr{B}} = \boldsymbol{T}$ iff $[\![s]\!]_{\mathscr{B}} = \boldsymbol{F}$ iff $\mathscr{B}(s) = \boldsymbol{F}$ iff $\neg s$ occurs on $\boldsymbol{B}$; which of course it does.

Induction step: Suppose β is a complex sentence, and the lemma holds of its less complex constituents. There are three cases of interest, corresponding to different tableau rules: β is a double negation; a branch rule can be applied to β; or a list rule can be applied to β.

1. β is a double negation, $\neg\neg\phi$. Then, because this is a finished branch, ϕ appears on β; because the lemma holds of less complex constituents, $[\![\phi]\!]_{\mathscr{B}} = \boldsymbol{T}$. So $[\![\neg\phi]\!]_{\mathscr{B}} = \boldsymbol{F}$, and $[\![\neg\neg\phi]\!]_{\mathscr{B}} = \boldsymbol{T}$, i.e., $[\![\beta]\!]_{\mathscr{B}} = \boldsymbol{T}$.

2. β can have a list rule applied to it. Let us choose Negated Conditional, so $\beta = \neg(\phi \to \psi)$. Then ϕ and $\neg\psi$ appear on $\boldsymbol{B}$, because the tableau is finished, and by the induction hypothesis $[\![\phi]\!]_{\mathscr{B}} = [\![\neg\psi]\!]_{\mathscr{B}} = \boldsymbol{T}$. Therefore by the rules on the valuation function, $[\![\phi \to \psi]\!]_{\mathscr{B}} = \boldsymbol{F}$, and $[\![\neg(\phi \to \psi)]\!]_{\mathscr{B}} = \boldsymbol{T}$, i.e., $[\![\beta]\!]_{\mathscr{B}} = \boldsymbol{T}$, as required. Analogous reasoning applies to the other list rules.

3. β can have a branch rule applied to it. Let $\beta = \phi \vee \psi$. Then either ϕ or ψ appears on $\boldsymbol{B}$; hence either $[\![\phi]\!]_{\mathscr{B}} = \boldsymbol{T}$ or $[\![\psi]\!]_{\mathscr{B}} = \boldsymbol{T}$, which by the rules on the valuation function, means that $[\![\phi \vee \psi]\!]_{\mathscr{B}} = \boldsymbol{T}$, i.e., $[\![\beta]\!]_{\mathscr{B}} = \boldsymbol{T}$, as required. Analogous reasoning applies to the other branch rules.

That suffices to show the lemma. □

Theorem 83 (Completeness for tautologies)
If $\models \phi$ *then* $\vdash \phi$.

Proof. We prove the equivalent contrapositive, namely, that if it is *not* the case that ϕ is a syntactic theorem (which we write '$\nvdash \phi$') then it is not the case that ϕ is a semantic theorem ('$\not\models \phi$').

Assume that $\nvdash \phi$. Then there is a finished open tableau generated by $\{\neg\phi\}$ which has an open branch, $\boldsymbol{B}$. Let $\mathscr{B}$ be the structure induced by $\boldsymbol{B}$, in accordance with Hintikka's Lemma (Lemma 82). By that lemma, every sentence occurring on $\boldsymbol{B}$ is true in $\mathscr{B}$. Since $\neg\phi$ appears on $\boldsymbol{B}$, as the root, $[\![\neg\phi]\!]_{\mathscr{B}} = 1$. Hence there is a structure which makes ϕ false, so $\not\models \phi$. □

Theorem 83 states that every tautology is derivable. This can easily be extended to arguments with finitely many premises:

Theorem 84 (Completeness for finite arguments)
When Γ *is finite, if* $\Gamma \models \phi$ *then* $\Gamma \vdash \phi$.

Proof. If $\Gamma \models \phi$, and Γ is a finite set $\{\gamma_1, \ldots, \gamma_n\}$, then (by repeated applications of the Deduction theorem for entailment (Theorem 33), $(\gamma_1 \to (\gamma_2 \to (\ldots(\gamma_n \to \phi)\ldots)))$ is a tautology. By Theorem 83, $\vdash (\gamma_1 \to (\gamma_2 \to (\ldots(\gamma_n \to \phi)\ldots)))$. By repeated applications of the deduction theorem for tableaux (Theorem 71), $\Gamma \vdash \phi$. □

Completeness for Arbitrary Arguments Theorem 84 shows that if an argument with finitely many premises is valid, there is a tableau derivation of the conclusion from the premises. But what if there are infinitely many premises in Γ? Could a claim validly follow from infinitely many premises, and yet there be no tableau derivation associated with that argument?

We already have enough materials to see that the answer is no.

Theorem 85 (Completeness from Compactness)
For any set of $\mathscr{L}_1$ *sentences* Γ*, if* $\Gamma \models \phi$ *then* $\Gamma \vdash \phi$.

Proof. If $\Gamma \models \phi$, then $\Gamma \cup \{\neg\phi\}$ is unsatisfiable. By Compactness (Theorem 60), if $\Gamma \cup \{\neg\phi\}$ is unsatisfiable, then for some finite set $\Sigma \subseteq \Gamma$, $\Sigma \cup \{\neg\phi\}$ is unsatisfiable, i.e., $\Sigma \models$. Accordingly, $\Sigma \models \phi$. By Theorem 84, $\Sigma \vdash \phi$; and since $\Sigma \subseteq \Gamma$, $\Gamma \vdash \phi$. □

We can also prove completeness directly without detouring through compactness. There are no infinite tableaux, so we need to pursue our goal with some subtlety. Rather than consider a tree generated by the entire infinite set Γ, we will consider a sequence of finite tableaux, each of which is generated by successively more of Γ, and each of which contains the previous tableau as a proper part.

Theorem 86 (Completeness Directly)
For any set of $\mathscr{L}_1$ sentences Γ, if $\Gamma \vDash \phi$ then $\Gamma \vdash \phi$.

Proof. Let Γ be enumerated $\gamma_1, \ldots$. Let $\mathbf{T}_0$ be a finished tableau generated by $\{\neg\phi\}$, and let each $\mathbf{T}_n$ be a finished tableau generated by $\langle \gamma_n, \ldots, \gamma_1, \neg\phi \rangle$, where $\mathbf{T}_n$ is constructed from $\mathbf{T}_{n-1}$ by adding γ_n as the topmost node, and applying any applicable tableau rule to it by extending any branch in $\mathbf{T}_{n-1}$ at the leaf, and then finishing the resulting tableau by extending that leaf to a tip by application of tableau rules. Note immediately that $\mathbf{T}_i$ is a proper part of $\mathbf{T}_{i+1}$; and if any branch is closed in $\mathbf{T}_i$, every branch including it in $\mathbf{T}_{i+1}$ remains closed.

If any $\mathbf{T}_i$ closes, then we can apply (84) to conclude that $\Gamma \vdash \phi$, since it is generated by a finite subset of $\Gamma \cup \{\neg\phi\}$. So let us assume for *reductio* that every $\mathbf{T}_i$ is open. Since closed branches never reopen, the fact that every tableau in our sequence is open means that each tableau contains at least one open branch that is contained within an open branch on all later tableaux in the sequence. By Hintikka's Lemma 82, for each $\mathbf{T}_i$ there is an $\mathscr{L}_1$ structure $\mathscr{B}_i$ that makes each member of $\{\neg\phi, \gamma_1, \ldots, \gamma_i\}$ true. Because each subsequent tableau retains at least one branch that is open on its predecessor, we know more specifically that this $\mathscr{B}_i$ can be chosen in such as way that it also makes true γ_{i+1}. And this is true for every $\mathbf{T}_i$, so in fact there is a single structure $\mathscr{B}$ such that each $\mathbf{T}_i$ has an open branch all of whose members are true in $\mathscr{B}$.

Since $\Gamma \cup \{\neg\phi\}$ is unsatisfiable, however, at least one $\gamma_j \in \Gamma$ is false in $\mathscr{B}$. But γ_j occurs in every open branch on $\mathbf{T}_j$, including the branch all of whose members are true in $\mathscr{B}$. Contradiction: not all of the $\mathbf{T}_i$s can be open. □

This in effect amounts to another proof of compactness.

Corollary 87 (Compactness from Completeness)
If $\Gamma \vDash$ then for some finite subset Σ of Γ, $\Sigma \vDash$.

Proof. If $\Gamma \models$, then by Theorem 86, $\Gamma \vdash$. By the definition of '$\vdash$' in Definition 98, there is a finished closed tableau generated by some finite subset Σ of Γ. By Soundness (Theorem 80), Σ is unsatisfiable. □

The proofs of Soundness and Completeness together show that whatever we can show using the semantic notions of satisfaction and truth in a structure, we could have equally well demonstrated by using the purely mechanical device of tableau derivation. A few simple rules for manipulating and deriving consequences from a collection of $\mathscr{L}_1$ sentences, simple enough that you could teach them to a child or a computer, nevertheless turn out to be powerful enough to demonstrate the correctness of every valid argument (and the incorrectness of every invalid argument) in propositional logic. This might not seem all that surprising – after all, it is fairly transparent to see that the rules governing tableaux were precisely chosen to have the right semantic properties. (They were not plucked randomly out of thin air.) But look again: isn't it at least a little remarkable that notions of truth and meaning can have their practical significance (in sorting arguments into conclusive and non-conclusive) captured entirely by formal rules which depend only on syntax? This is yet another sense in which our logic is formal: semantic consequence is extensionally equivalent to formal syntactic derivability.

7.6 Finitude and Decidability

In chapter 5 we considered issues around decidability by considering an effective procedure for checking validity using truth tables. We can use tableau derivations in an effective procedure too. In the previous chapter we showed that any finite set of $\mathscr{L}_1$ sentences generates a finished finite tableau (Theorem 66), and the procedure described in the proof for generating a finished tableau from a finite generating set is an effective one. So we can certainly produce finished tableau in a finite time.

Effective Evaluation of Tableaux For a decision procedure, however, we need to do more than produce a finished tableau – we need to check whether that tableau is closed. But since every $\mathscr{L}_1$ tree is finite, comprised of finitely many branches,

we can effectively check by exhaustive search whether it is closed. Enumerate the branches in some order; take the first branch, then for each node ϕ on the branch, check whether each branch descendent ψ to see whether ϕ is of the form $\neg\psi$ or ψ is of the form $\neg\phi$. If all the branches are closed, the tableau is closed. But we only needed to check finitely many nodes on finitely many branches, so we can perform this check in finite time.

A Tableau Decision Procedure Because a closed tableau will remain closed when finished, an unfinished closed tableau still shows the generating set to be inconsistent. This suggests the following decision procedure for inconsistency of a finite set Γ of $\mathscr{L}_1$ sentences:

1. Enumerate $\Gamma = \gamma_1, \ldots, \gamma_n$, and inscribe a tableau $\mathbf{T}_0$ which contains $\langle \gamma_1, \ldots, \gamma_n \rangle$ as its only branch.

2. Check whether $\mathbf{T}_0$ is closed; if it is, terminate with the report that Γ is inconsistent; otherwise continue to step 3.

3. If $\mathbf{T}_i$ is the last tableau constructed, then apply the tableau rules to γ_{i+1}, and then to any sentences resulting from γ_{i+1}, and to sentences resulting from them, etc. (Since each $\mathscr{L}_1$ sentence is of finite complexity, this process involves the generation of only finitely many sentences before the tableau rules no longer apply. The key is that after this process the only sentences which have not had tableau rules applied to them are original members of Γ.)

4. Apply the check for closure to $\mathbf{T}_{i+1}$. If it is closed terminate with the report that Γ is inconsistent; if it is open, and $\gamma_{i+2} \in \Gamma$, then return to step 3; if is is open and each $\gamma \in \Gamma$ has already had tableau rules applied to it, terminate with the report that Γ is consistent.

This procedure is a decision procedure for inconsistency: it terminates in a finite time after the construction of a finished tableau, and reports either open or closed. Because of the soundness and completeness theorems, this is also another decision procedure for unsatisfiability of finite sets of $\mathscr{L}_1$ sentences (Theorem 62).

Positive Decidability for Infinite Inconsistency We've already shown that there is no decision procedure as to whether or not an infinite set of $\mathscr{L}_1$ sentences is unsatisfiable. Because of soundness and completeness, if the question of consistency of an infinite set of sentences were decidable, we could use that procedure to construct a decision procedure for satisfiability. So consistency must also be undecidable.

But, just as earlier (Theorem 63), the question of inconsistency is positively decidable: if a recursively enumerable set of sentences is inconsistent, we can describe an effective procedure demonstrating that. The procedure is to begin with our enumeration of Γ, $\gamma_1, \ldots$. At stage i, take the set $\Gamma_i = \{\gamma_1, \ldots, \gamma_i\}$ to be the generating sequence for a tableau $\mathbf{T}_{\gamma_i}$; apply the procedure described above to check whether Γ_i is inconsistent. If it is, terminate with the message that Γ is inconsistent. If not, move on to stage $i+1$. It can readily be seen that this procedure will not terminate if Γ is consistent: the search for inconsistency will continue, constructing larger and larger open tableaux at each stage.

7.7 Trees and Tableaux

In this section some general theorems about trees are proved that are useful for situating the metatheory of tableaux in its mathematical context.

The Set-theoretic Conception of Tree The more general definition of a tree is set-theoretic:

Definition 103 (General Tree). A tree $\mathbf{T}$ is a pair $\langle N, \preceq \rangle$ where N is a set of items ('nodes'), and $\preceq$ is a partial order on N, such that (i) there is a *root node* $r \in N$ such that for all $n \in N$, $r \preceq n$, and (ii) for each $n \in N$, the set $\{x \in N : x \preceq n\}$ is well-ordered by $\preceq$ (recall Definition 33).

The nodes in this definition are otherwise featureless indices, individuated entirely by their place in the tree structure. In $\mathscr{L}_1$-trees, a given sentence can occur more than once on a branch, but a node cannot occur more than once in any tree. So if we want to show that $\mathscr{L}_1$-trees from Definition 90 are trees in the sense of Definition 103, we will have to come up with some way of associating $\mathscr{L}_1$ sentences with nodes. The simplest way is to *label* the tree. A labelled tree is a tree

together with a total function f from N into the set of $\mathscr{L}_1$ sentences, which assigns to each node a sentence as label. The function is into, so a given sentence might label more than one node.

Theorem 88 ($\mathscr{L}_1$-trees are labelled trees)
Each $\mathscr{L}_1$-tree corresponds to a labelled tree.

Proof. We want to show that, if we have an $\mathscr{L}_1$-tree $\mathbf{T}$, there is a corresponding tree $\mathbf{T}'$. So we need to show (1) that there is an appropriate set of nodes; (2) there is a partial order; (3) there is a root node; (4) the branches are well-ordered; and (5) there is a labelling function.

1. If X is a sequence $\langle x_1, \ldots, x_n, \ldots\rangle$, define an *index set* X_n as follows: $X_n = \{\langle x_i, i\rangle : x_i \in X\}$. This transforms an ordered sequence of items into an unordered set of item, position pairs.

   Recalling that an $\mathscr{L}_1$-branch is just a sequence of $\mathscr{L}_1$ sentences, and an $\mathscr{L}_1$-tree is just a set of $\mathscr{L}_1$-branches, there will be an index set for each $\mathscr{L}_1$-branch in $\mathbf{T}$. The union of all index sets for $\mathscr{L}_1$-branches in our $\mathscr{L}_1$-tree $\mathbf{T}$ will be the set N of nodes for our desired tree $\mathbf{T}'$. Each ⟨sentence, index⟩ pair occuring on some branch is included. Because sets with the same members are identical, and when branches overlap they have the same sentences at the same position, then we do not have redundancy: though every $\mathscr{L}_1$-branch shares the root node, the pair $\langle\rho, 1\rangle$ will be added to the union of all index sets 'several times', but of course we end up with just one member of the set for that shared node. Since we have said nothing about what the set of nodes is to consist in, it is perfectly acceptable for this complex set of pairs to be the set of nodes for $\mathbf{T}'$.

2. We need to define the partial order $\preceq$ on N. Since each member of N is of the form $\langle\phi, i\rangle$, it suffices to define it on pairs of this form. We will say that $\langle\phi, i\rangle \preceq \langle\psi, n\rangle$ iff there is an $\mathscr{L}_1$-branch on $\mathbf{T}$ that includes ϕ at position i and ψ at position j, and $i \leq j$. So defined, $\preceq$ is a partial order (Definition 32): it is transitive (since if $x \preceq y$ and $y \preceq z$, then all of x, y, z fall on the same $\mathscr{L}_1$-branch, and the transitivity of $\leq$ ensures that $x \preceq z$); it is reflexive (since $\leq$ is reflexive; and it is antisymmetric (since if $x \preceq y$ and $y \preceq x$, then x and

y fall on the same $\mathscr{L}_1$-branch and $i \leq j$ and $j \leq i$, which entails that $i = j$, and since there is only one sentence at each position in a branch, $x = y$).

3. Since every $\mathscr{L}_1$-branch on $\mathbf{T}$ shares a common initial subsequence, each begins with some sentence ϕ. So $\langle \phi, 1 \rangle$ is the root node of $\mathbf{T}'$.

4. Recall that a well-ordering is a total order on a set such that every non-empty subset has a least member – there are no infinite descending chains. We need to show that $\preceq$, as defined, when restricted to sets of the form $\{x \in N : x \preceq n\}$ for some $n \in N$, is a well-ordering. Since $x \preceq y$ only if x and y fall on the same $\mathscr{L}_1$-branch, the sets of this form are index sets of initial subsequences of $\mathscr{L}_1$-branches. Restricted to these intial subsequences, $\preceq$ is total because $\leq$ is total; and since the initial subsequences are sequences, they can be mapped one-one onto subsets of $\mathbb{N}$, and the least number principle then entails that $\leq$ is a well-ordering on these initial subsequences, so that $\preceq$ is a well-ordering on the corresponding index sets.

5. The labelling function is: $f(\langle \phi, i \rangle) = \phi$. This simply labels a node with its first element, which is the sentence from the original $\mathscr{L}_1$-tree. □

Infinite Trees The set-theoretic account of trees permits trees with infinitely many nodes. Such a tree might be infinitely tall, with an infinite branch descending from the root; or it might be infinitely wide, with infinitely many branches. But if there are only finitely many branches from each node, an infinite tree must have an infinite branch.

Lemma 89 (König's Lemma for Trees)
Every finitely-branching tree containing infinitely many sentences has an infinite branch.

Proof. (See also Beall and van Fraassen (2003: 152).) Let us say that ψ is a *descendent* of ϕ iff there is a branch on which both occur and where ψ occurs after ϕ in the sequence. If a tree contains infinitely many sentences, the root node v_0 must have infinitely many descendents.

Suppose a node v_n on an infinite tree has infinitely many descendents. Then there is a node v_{n+1} immediately posterior to and on the same branch as v_n with

infinitely many descendents. For if every branch $\boldsymbol{B}$ on which ν_n occurs is such that the next item in the sequence after ν_n has only finitely many descendents, then every branch on which ν_n occurs must be finite. Since ν_n has infinitely many descendents, there must be infinitely many branches $\boldsymbol{B}_i$ agreeing with $\boldsymbol{B}$ up through ν_n but disagreeing on the next item in the sequence: $\boldsymbol{B}_i = \langle \rho, \ldots, \nu_n, \phi_i, \ldots, \lambda \rangle$. But this cannot be a tree, since for any finitely branching tree, there are at most finitely many distinct branches agreeing with $\boldsymbol{B}$ up through ν_n. So at least one immediate descendent of ν_n must have infinitely many descendents; pick one and call it ν_{n+1}.

We have shown that the root ν_0 has infinitely many descendents, and that if a node ν_n has infinitely many descendents, then there is an immediate descendent ν_{n+1} which also has infinitely many descendents. The sequence $\langle \nu_0, \ldots, \nu_n, \ldots \rangle$ is an infinite sequence of $\mathscr{L}_1$ sentences, each of which follows from previous members of the sequence by the tree rules. So it is an infinite tree branch. □

We used this lemma in the proof of the finitude of tableaux (Theorem 66), where the finitude of finished $\mathscr{L}_1$ tableau branches was used to establish the finitude of $\mathscr{L}_1$ tableaux.

Further Reading

Smullyan (1968), Beall and van Fraassen (2003), Hodges (2001), Bostock (1997: ch. 4) and Jeffrey (2006) all discuss the metatheory of tableaux systems in ways that may be illuminating to consider in comparison to the results and discussion offered above.

Exercises

1. Show Weakening for $\vdash$: that if $\Gamma \vdash \phi$ then $\Gamma \cup \Delta \vdash \phi$ for any Δ.
2. Prove that Theorem 77 entails Theorem 76, thus showing that those two theorems are equivalent.
3. Assuming the soundness theorem, prove that if $\Gamma \vDash \phi$ is an *incorrect* sequent then $\Gamma \vdash \phi$ is also incorrect.
4. Consider an operator $\odot$ defined by the following tableau rules:

$$\begin{array}{c} (\phi \odot \psi) \\ | \\ (\phi \odot \phi) \\ (\psi \odot \psi) \end{array} \qquad\qquad \begin{array}{c} ((\phi \odot \psi) \odot (\phi \odot \psi)) \\ \diagup \quad \diagdown \\ \phi \qquad \psi \end{array}$$

(a) Which truth-function does $\odot$ express? (I.e., which truth function is characterised by these tableau rules?)

(b) If these rules were the only rules for a system of tableau, under what conditions should a branch be counted as closed?

(c) Are these tableau rules sound?

5. A fellow student argues as follows:

> If the derivation system just introduced is complete, then whenever $\Gamma \vDash \phi$, there is a derivation that shows $\Gamma \vdash \phi$. But proofs are finite; so if Γ is infinite, there is no proof that $\Gamma \vdash \phi$ – so the system is not complete!

How should you respond?

Answers to selected exercises on page 250.

Chapter 8

Natural Deduction Derivations in $\mathscr{L}_1$; Soundness and Completeness

Inferentialism A less sceptical take on any mooted correspondence between formal arguments which follow good rules and valid arguments is that the formal rules do in fact provide meanings for the expressions of the language. Rather than simply setting down the meanings of expressions by stipulation, or despairing about how there could be any facts about meaning at all, we might think meaning emerges from the underlying syntax.

This inferentialism……XXX

8.1 Natural Deduction Proofs

Proof In the last chapter, we saw that using truth tables was a decidable method for evaluating sequents, but it is cumbersome and inelegant. Here, we consider elegant and powerful *proof systems* for sentential logic.

A *proof* in the informal sense is something which establishes the truth of some claim. Our proofs are structures made out of sentences of $\mathscr{L}_1$, with the following

desirable properties:

- Each step in the proof involves an obviously correct rule.
- Whenever the earlier sentences are true, the later sentences will be also; so the terminal conclusion of a proof should always be true whenever the assumptions are.
- All the logical truths should have a proof.

In this chapter, we'll see that there do exist proofs that meet these conditions for $\mathscr{L}_1$; and thus that we can establish various claims in $\mathscr{L}_1$ without needing to consider all the $\mathscr{L}_1$-structures.

Natural Deduction: Assumptions and $\vdash$ The first proof system is the *natural deduction* system introduced in The Logic Manual.

In this system, every proof begins with *assumptions. Any sentence may be an assumption.* A proof terminates with a *conclusion.*

Whenever there is a correctly constructed proof that ϕ, on the basis of assumptions $\gamma_1, \ldots, \gamma_n$, we write this *syntactic sequent*:

$$\gamma_1, \ldots, \gamma_n \vdash \phi.$$

This is to be read 'ϕ is *provable from* $\gamma_1, \ldots, \gamma_n$', or '$\gamma_1, \ldots, \gamma_n$ *syntactically entail(s)* ϕ'. If ϕ can be proved with no assumptions at all, we write $\vdash \phi$; in that case, we say that ϕ is a *theorem.*

Any sentence ϕ may be assumed; applying no rules at all, the proof therefore terminates with ϕ. So $\phi \vdash \phi$; not perhaps the most interesting result, but nevertheless intuitively correct.

Natural Deduction Rules As any assumption is a proof on its own, our rules are rules for constructing new proofs from existing proofs, often (but not always) retaining the assumptions of the existing proofs.

Some rules permit an assumption to be *discharged.* Suppose you have a proof of ϕ on the basis of assumption γ. You thus show that *if* you had a proof of γ, *then* you could construct a proof of ϕ; and this on its own is a proof of $\gamma \to \phi$,

$$\frac{\begin{array}{c}[\gamma]\\ \vdots\\ \phi\end{array}}{\gamma \to \phi}\ {\to}\text{Intro} \qquad \frac{\gamma \to \phi \qquad \gamma}{\phi}\ {\to}\text{Elim}$$

$$\frac{\begin{array}{cc}[\phi] & [\phi]\\ \vdots & \vdots\\ \psi & \neg\psi\end{array}}{\neg\phi}\ \neg\text{Intro} \qquad \frac{\begin{array}{cc}[\neg\phi] & [\neg\phi]\\ \vdots & \vdots\\ \psi & \neg\psi\end{array}}{\phi}\ \neg\text{Elim}$$

Table 8.1: Natural Deduction Rules for $\mathscr{L}_{\neg,\to}$

whether or not γ is actually true. The assumption γ has been discharged. Note that γ needn't be used, or even occur in the proof, to be discharged: the rules entitle one to discharge *every* occurence of a sentence – even if there are none!

We can consider two types of obvious rules: those which construct a proof of a new sentence of less complexity than sentence proved by the existing proof(s), and those which construct a proof of a sentence of increased complexity. We call the first type of rule *elimination* rules, and the second type *introduction* rules.

Natural Deduction Rules for $\mathscr{L}_{\neg,\to}$ Let us consider for the moment the fragment of $\mathscr{L}_1$ involving only $\neg$ and $\to$, $\mathscr{L}_{\neg,\to}$. We have in this fragment four rules, laid out in Table 8.1, elimination rules and introduction rules for each connective. I write $[\gamma]$ to show that the rule permits the assumption γ to be discharged. Call this system of rules, plus the rules about assumptions, $ND_{\neg,\to}$.

Deduction Theorem for $ND_{\neg,\to}$

Theorem 90 (Deduction Theorem)
$\Gamma \vdash \psi \to \phi$ *iff* $\Gamma, \psi \vdash \phi$.

Suppose $\Gamma, \psi \vdash \phi$. Then we can apply $\to$Intro as follows:

$$\frac{\begin{array}{c}\Gamma, [\psi]\\ \vdots\\ \phi\end{array}}{\psi \to \phi}\ {\to}\text{Intro}$$

Therefore, $\Gamma \vdash \psi \to \phi$. Similarly, suppose $\Gamma \vdash \psi \to \phi$:

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ \psi \to \phi\end{array} \qquad \psi}{\phi} \to\text{Elim}$$

That is, $\Gamma, \psi \vdash \phi$.

¬Elim can be replaced by ¬¬Elim Consider the rule:

$$\dfrac{\neg\neg\phi}{\phi} \neg\neg\text{Elim}$$

This rule can obviously be derived from our rules: the correctness of the syntactic sequent $\neg\neg\phi \vdash \phi$ is demonstrated by this proof:

$$\dfrac{\neg\neg\phi \quad [\neg\phi]}{\phi} \neg\text{Elim}$$

But with ¬¬Elim we can show the rule ¬Elim is redundant, using ¬Intro:

$$\dfrac{\dfrac{\begin{array}{c}[\neg\phi] \\ \vdots \\ \psi\end{array} \qquad \begin{array}{c}[\neg\phi] \\ \vdots \\ \neg\psi\end{array}}{\neg\neg\phi} \neg\text{Intro}}{\phi} \neg\neg\text{Elim}$$

8.2 A Little More Proof Theory

'Proof theory' is the mathematical examination of proofs conceived of as mathematical objects in their own right. In this section we explore a little of what this involves. We've already seen one some proof theory in section 1 of this chapter, where we showed the redundancy of the proof rule ¬Elim in the presence of ¬¬Elim, and vice versa.

The Language $\mathscr{L}_{DP}$ Consider the language $\mathscr{L}_{DP}$ which contains all sentence letters, and all sentences involving only the connectives $\wedge, \to$, and also contains a

$$\dfrac{\begin{array}{cc}[\phi] & [\phi] \\ \vdots & \vdots \\ \psi & \psi \to \bot\end{array}}{\phi \to \bot}\ \neg\text{Intro}_{DP} \qquad \dfrac{\begin{array}{cc}[\phi \to \bot] & [\phi \to \bot] \\ \vdots & \vdots \\ \psi & \psi \to \bot\end{array}}{\phi}\ \neg\text{Elim}_{DP}$$

Table 8.2: Negation rules in $\mathscr{L}_{DP}$

sentential constant $\bot$. A sentential constant is syntactically like a sentence letter, but (unlike sentence letters) it gets a constant interpretation in every structure – in this case, for any structure $\mathscr{A}$, $|\bot|_{\mathscr{A}} = F$. The atomic sentences of $\mathscr{L}_{DP}$ thus include $\bot$ and all sentence letters; complex sentences are then built up in the usual way.

Translating between $\mathscr{L}_{DP}$ and $\mathscr{L}_1$ It's clear that the $\mathscr{L}_1$ sentence '$\neg\phi$' expresses the same truth function as the $\mathscr{L}_{DP}$ sentence '$\phi \to \bot$' (consideration of a truth table will suffice to show this). Using this translation, we can give $\mathscr{L}_{DP}$-versions of the natural deduction rules for $\neg$ in $\mathscr{L}_1$, as seen in Table 8.2.

With the negation rules as specified, we can prove this:

Theorem 91 (Alternative Negation Rules for $\mathscr{L}_{DP}$)
Π *is a proof of some sequent in the language* $\mathscr{L}_{DP}$ *that makes a use* u *of one of the Table 8.2 versions of Halbach's rules iff there is another proof* Π' *of the same sequent which replaces that use* u *by a construction which only uses this rule for* $\neg$ *(together with the rules for* $\to$*):*

$$\dfrac{\begin{array}{c}[\phi \to \bot] \\ \vdots \\ \bot\end{array}}{\phi}\ \bot_C$$

Proof. If: There are two cases. Case 1: Suppose Π is a proof that makes use of $\neg\text{Intro}_{DP}$. Then we can replace that use by the following construction, which

yields another proof Π' which uses only the rules for $\rightarrow$:

$$\dfrac{\dfrac{\begin{array}{c}[\phi]\\ \vdots\\ \psi\end{array} \qquad \begin{array}{c}[\phi]\\ \vdots\\ \psi \rightarrow \bot\end{array}}{\bot}\rightarrow \text{Elim}}{\phi \rightarrow \bot}\rightarrow \text{Intro}$$

Case 2: Suppose Π is a proof that makes use of $\neg\text{Elim}_{DP}$. Then we can replace that use by the following construction, which yields another proof Π' which uses only the new rule $\bot_C$ and the rule $\rightarrow$Elim:

$$\dfrac{\dfrac{\begin{array}{c}[\phi \rightarrow \bot]\\ \vdots\\ \psi\end{array} \qquad \begin{array}{c}[\phi \rightarrow \bot]\\ \vdots\\ \psi \rightarrow \bot\end{array}}{\bot}\rightarrow \text{Elim}}{\phi}\bot_C$$

Only if: Suppose Π is a proof which makes use of the new rule $\bot_C$. Then we may replace that use by the following construction which uses only the old rules for $\rightarrow$ and $\neg\text{Elim}_{DP}$.

□

$$\dfrac{\begin{array}{c}[\phi \rightarrow \bot]\\ \vdots\\ \bot\end{array} \qquad \dfrac{[\bot]}{\bot \rightarrow \bot}\rightarrow \text{Intro}}{\phi}\neg\text{Elim}_{DP}$$

Reducing Complexity of Proofs Return now to the $\mathscr{L}_1$-fragment containing only $\rightarrow, \wedge, \neg$. Let the *(degree of) complication* of a sentence be the number of connectives occurring in the sentence (a sentence letter has degree of complication 0, if ϕ and ψ have degree of complication m and n respectively, then $(\phi \wedge \psi)$ has degree of complication $m + n + 1$, etc.) We can now show:

Theorem 92 (Reducing Complexity of Proofs)
If Π *is a proof in Halbach's system for* $\mathscr{L}_{\rightarrow,\wedge,\neg}$ *that* $\Gamma \vdash \phi$*, in which the most complicated sentence resulting from an application of* $\neg$*Elim is* ψ*, then (when* ψ *is not atomic) there is a distinct proof* Π' *of* $\Gamma \vdash \phi$ *in which the results of all applications of*

¬Elim are less complicated *than ψ. Indeed, there exists a proof $\Pi^\dagger$ of that sequent, in which the result of every application of ¬Elim is atomic.*

Proof. Suppose that the most complicated result of ¬Elim in Π is ψ. So part of Π looks like this:

$$\dfrac{\Gamma \quad \begin{array}{c}[\neg\psi]\\ \vdots\\ \chi\end{array} \quad \Delta \quad \begin{array}{c}[\neg\psi]\\ \vdots\\ \neg\chi\end{array}}{\psi}\ \neg\text{Elim}$$

ψ is complicated (not a sentence letter), so given there are three connectives in this language, there are three cases:

1. $\psi = \neg\pi$. In which case replace that part of the proof sketched above with this one:

$$\dfrac{\Gamma \quad \begin{array}{c}\dfrac{[\pi] \quad [\neg\pi]}{\neg\neg\pi}\ \neg\text{Intro}\\ \vdots\\ \chi\end{array} \quad \Delta \quad \begin{array}{c}\dfrac{[\pi] \quad [\neg\pi]}{\neg\neg\pi}\ \neg\text{Intro}\\ \vdots\\ \neg\chi\end{array}}{\neg\pi}\ \neg\text{Intro}$$

2. $\psi = \pi \wedge \xi$. Proof left for Exercise.

3. $\psi = \pi \to \xi$. In which case replace that part of the proof sketched above with this one:

$$\dfrac{\Gamma \quad \begin{array}{c}\dfrac{\dfrac{[\pi \to \xi] \quad [\pi]}{\xi}\ \to\text{Elim} \quad [\neg\xi]}{\neg(\pi \to \xi)}\ \neg\text{Intro}\\ \vdots\\ \chi\end{array} \quad \Delta \quad \begin{array}{c}\dfrac{\dfrac{[\pi \to \xi] \quad [\pi]}{\xi}\ \to\text{Elim} \quad [\neg\xi]}{\neg(\pi \to \xi)}\ \neg\text{Intro}\\ \vdots\\ \neg\chi\end{array}}{\dfrac{\xi}{\pi \to \xi}\ \to\text{Intro}}\ \neg\text{Elim}$$

Notice that applying these replacements repeatedly to uses of ¬Elim in a proof will eventually replace all complex sentences in uses of ¬Elim by simpler ones. All

sentences of $\mathscr{L}_1$ are finitely complex, and all proofs involve only finitely many applications of a given rule, this process terminates after a finite time with only atomic sentences as the conclusions of uses of ¬Elim. □

It is clear that we can put the results of Theorems 91 and 92 together, to establish that, in the system which involves the normal rules for $\wedge$ and $\rightarrow$, and has $\bot_C$ as its only other rule, if Π is a proof in the system DP that $\Gamma \vdash_{DP} \phi$, then there exists a proof of that sequent, $\Pi^\dagger$, in which the result of any application of $\bot_C$ is atomic. (I leave the proof as a problem.)

Further Reading

Natural deduction was invented by Gentzen (1969), the founder of proof theory. The proof theory of natural deduction systems discussed in this chapter is based on Prawitz (2006: 39–41); he uses Theorems 91 and 92, and the last problem, and proves the famous 'cut elimination' theorem for natural deduction, unfortunately a topic beyond the scope of these notes.

The axiom system here was developed by Jan Łukasiewicz, simplifying Frege's system: see Borkowski and Słupecki (1958: p. 25). (The Polish notation used in this article is initially confusing, but has the wonderful feature that it is entirely unambiguous without the use of parentheses.) See also Bostock (1997: ch. 5–6); he gives a direct proof of the completeness of a system based on Ł at pp. 217–9.

Exercises

1. (a) Devise introduction and elimination rules for a natural deduction system for the language $\mathscr{L}_\uparrow$ with the Sheffer Stroke as its only connective, being sure to fully justify your answer. How many such rules do we need?

   (b) Suppose in a system of natural deduction ND for $\mathscr{L}_1$ we replaced the ¬Elim rule by the following, *ex falso quodlibet*:

   $$\frac{\psi \qquad \neg\psi}{\phi}\ \text{EFQ}$$

   i. Is the resulting system ND' equivalent (in terms of what can be proved) to the original system? (Justify your answer; it is somewhat difficult to prove conclusively, but give some reasons for your view.)

ii. What happens if we, in addition, replace ¬Intro by the following rule, *tertium non datur*, yielding the system ND''?

$$\dfrac{\begin{array}{cc}[\phi] & [\neg\phi] \\ \vdots & \vdots \\ \psi & \psi\end{array}}{\psi}\ \text{TND}$$

(c) Consider the natural deduction system $ND_{\to,\vee}$ for the language $\mathscr{L}_{\to,\vee}$ in which the only connectives are $\to, \vee$. Suppose we introduce to the language the zero-place sentential constant $\bot$, and add the following rules to the proof system, yielding $ND_{\to,\vee,\bot}$:

$$\dfrac{\dfrac{\bot}{\phi}}{\bot}\ \bot \qquad \dfrac{\begin{array}{c}[\phi \to \bot] \\ \vdots \\ \bot\end{array}}{\phi}\ \text{C}$$

i. Give a natural translation between $\mathscr{L}_{\to,\vee,\bot}$ and $\mathscr{L}_{\to,\vee,\neg}$.

ii. Using that translation, can you show that the proof system $ND_{\to,\vee,\bot}$ just introduced is equivalent to the natural deduction system $ND_{\to,\vee,\neg}$ involving just Halbach's rules for $\to$, $\vee$ and $\neg$?

(d) A mystery connective $\oplus$ has the following introduction and elimination rules:

$$\dfrac{\phi \oplus \psi}{\psi}\ \oplus\text{Elim} \qquad \dfrac{\phi}{\phi \oplus \psi}\ \oplus\text{Intro}$$

i. Show that no system containing a connective defined by these rules is sound.

ii. What limits should be placed on the ability of introduction and elimination rules to define or characterise the inferential role of a connective in light on this result? (This question invites discussion, not a definitive answer. You may wish to consult Prior (1960).)

(e) Relatedly to 1.(d).ii:

i. Check, by means of truth tables, that $(((P \to Q) \to P) \to P)$ is a tautology.

ii. Give a natural deduction proof of $(((P \to Q) \to P) \to P)$.

iii. Every valid argument involving only sentences containing $\wedge$ as their only connective can be proved valid using just the rules $\wedge$Intro and $\wedge$Elim. In that sense, $\wedge$ is completely characterised by its introduction and elimination rules. What do the previous results show about $\rightarrow$ in this connection?

2. Prove the omitted case (where the complex sentence is a conjunction) in the proof of Theorem 92.

3. Show that, in the system which involves the normal rules for $\wedge$ and $\rightarrow$, and has $\bot_C$ as its only other rule, if Π is a proof in the system DP that $\Gamma \vdash_{DP} \phi$, then there exists a proof of that sequent, $\Pi^\dagger$, in which the result of any application of $\bot_C$ is atomic.

Chapter 9

Natural Deduction Metatheory: Soundness and Completeness

9.1 Soundness for Natural Deduction

Soundness and Completeness A proof system P in a given language is *sound* with respect to a semantic interpretation of the language just in case whenever there is a proof which establishes $\Gamma \vdash_P \phi$, it is the case that $\Gamma \models \phi$.

A proof system P in a given language is *complete* with respect to a semantic interpretation of the language just in case whenever it is the case that $\Gamma \models \phi$, there is a proof which establishes $\Gamma \vdash_P \phi$.

In this section and the next, we will show that the natural deduction system $ND_{\neg,\rightarrow}$ just introduced is *sound and complete* for the standard semantics of $\mathscr{L}_{\neg,\rightarrow}$. Given the expressive adequacy of $\mathscr{L}_{\neg,\rightarrow}$, that also shows the completeness of the natural deduction system with respect to $\mathscr{L}_1$; to show the soundness of that system, we also need to show that the other rules of the system ND are also sound.

Soundness of $ND_{\neg,\rightarrow}$

Theorem 93

The Rules of $ND_{\neg,\rightarrow}$ are sound.

Proof. Suppose that $\Gamma \vdash \phi$. We show by induction on the complexity of proofs that $\Gamma \vDash \phi$.

Base case: The least complex proof is a single node, ϕ, establishing $\phi \vdash \phi$. In this case, any structure which assigns T to ϕ clearly assigns T to ϕ, so $\phi \vDash \phi$, i.e., $\Gamma \vDash \phi$.

Induction step: Suppose $\Gamma \vdash \phi$, established by applying one of the natural deduction rules to some previously obtained proofs to obtain a proof of ϕ. There are four cases:

→Elim From a proof of ψ on assumptions Δ, and a proof of $\psi \rightarrow \phi$ on assumptions Θ, obtain a proof of ϕ on assumption $\Gamma = \Delta \cup \Theta$. By the induction hypothesis, $\Delta \vDash \psi$ and $\Theta \vDash \psi \rightarrow \phi$, so every structure which makes all of the members of Γ true, makes ψ and $\psi \rightarrow \phi$ true, and therefore ϕ must be true in all such structures, showing that $\Gamma \vDash \phi$.

→Intro From a proof of ϕ on assumptions $\Gamma \cup \{\psi\}$, apply →Intro to obtain a proof of $\psi \rightarrow \phi$ on assumption Γ (discharging ψ). But since by hypothesis $\Gamma, \psi \vDash \phi$, by the deduction theorem, $\Gamma \vDash \psi \rightarrow \phi$.

¬Elim From a proof of ψ on assumptions $\Gamma, \neg\phi$ and a proof of $\neg\psi$ on assumptions $\Gamma, \neg\phi$, obtain a proof of ϕ on assumptions Γ, discharging $\neg\phi$. Since $\Gamma, \neg\phi \vDash \psi$ and $\Gamma, \neg\phi \vDash \neg\psi$, it follows that $\Gamma, \neg\phi \vDash$; therefore $\Gamma \vDash \phi$.

¬Intro From a proof of ψ on assumptions Γ, ϕ and a proof of $\neg\psi$ on assumptions Γ, ϕ, obtain a proof of $\neg\phi$ on assumptions Γ, discharging ϕ. Since $\Gamma, \phi \vDash \psi$ and $\Gamma, \phi \vDash \neg\psi$, it follows that $\Gamma, \phi \vDash$; therefore $\Gamma \vDash \neg\phi$.

Thus all the rules preserve soundness; no proof in $ND_{\neg,\rightarrow}$ can be constructed other than using these rules; so $ND_{\neg,\rightarrow}$ is sound. □

9.2 Completeness

Consistency A set of sentences Γ is called *(syntactically) consistent* iff $\Gamma \nvdash \neg(\phi \to \phi)$. Otherwise it is *inconsistent*, also written $\Gamma \vdash$. Equivalently, consider this proof:

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ \neg(\phi \to \phi)\end{array} \qquad \dfrac{[\phi]}{\phi \to \phi}\,{\to}\text{Intro}}{\psi}\,\neg\text{Elim}$$

If we have a proof of $\neg(\phi \to \phi)$, we have a proof of arbitrary ψ from the assumption Γ; so we can say that Γ is consistent iff there is a sentence *not provable* from Γ.

A set of sentences Γ is *maximally consistent* iff it is consistent and for any ϕ, if $\phi \notin \Gamma$, then $\Gamma, \phi \vdash$ (one cannot add any more sentences while remaining consistent).

Properties of Maximal Consistent Sets: Deductive Closure

Lemma 94 (Deductive Closure)
If Γ is a maximal consistent set, then if $\Gamma \vdash \phi$, then $\phi \in \Gamma$.

Proof. Suppose that $\Gamma \vdash \phi$, but $\phi \notin \Gamma$. Since Γ is maximal consistent, $\Gamma, \phi \vdash \neg(\phi \to \phi)$. Using $\to$Intro and discharging ϕ, we see that $\Gamma \vdash \phi \to (\neg(\phi \to \phi))$. But now use $\to$Elim, and obtain $\Gamma \vdash \neg(\phi \to \phi)$ i.e., Γ is inconsistent. □

Properties of Maximal Consistent Sets: Negation Completeness

Lemma 95 (Negation Completeness)
If Γ is a maximal consistent set, then $\neg\phi \in \Gamma$ iff $\phi \notin \Gamma$.

Proof. L to R: If both ϕ and $\neg\phi$ were in Γ, we could prove $\neg(\phi \to \phi)$ by an application of $\neg$Elim, showing Γ inconsistent.

R to L: If $\phi \notin \Gamma$, then $\Gamma, \phi \vdash \neg(\phi \to \phi)$:

$$\dfrac{\begin{array}{c}\Gamma, [\phi] \\ \vdots \\ \neg(\phi \to \phi)\end{array} \qquad \dfrac{[\phi]}{\phi \to \phi}\,{\to}\text{Intro}}{\neg\phi}\,\neg\text{Intro}$$

Therefore, $\Gamma \vdash \neg\phi$; by Deductive Closure Lemma, $\neg\phi \in \Gamma$. □

Properties of Maximal Consistent Sets: Conditional Completeness

Lemma 96 (Conditional Completeness)
If Γ is any maximal consistent set, then $\phi \to \psi \in \Gamma$ iff whenever $\phi \in \Gamma$, it follows that $\psi \in \Gamma$.

Proof. L to R: Suppose $\phi \to \psi \in \Gamma$. Then $\Gamma \vdash \phi \to \psi$; and if $\phi \in \Gamma$, then $\Gamma \vdash \phi$. Apply $\to$Elim, and establish $\Gamma \vdash \psi$, i.e., $\psi \in \Gamma$.

R to L: Suppose that whenever $\phi \in \Gamma$, $\psi \in \Gamma$. Consider two cases:

1. $\phi \in \Gamma$. So $\Gamma = \Gamma \cup \{\phi\}$. Moreover, $\psi \in \Gamma$, so $\Gamma, \phi \vdash \psi$. Apply $\to$Intro and discharge ϕ to show $\Gamma \vdash \phi \to \psi$.

2. $\phi \notin \Gamma$. Then $\Gamma, \phi \vdash \neg(\phi \to \phi)$. But then $\Gamma, \phi \vdash \psi$ (by a similar proof as just above); applying $\to$Intro and discharge ϕ, $\Gamma \vdash \phi \to \psi$. □

Properties of Maximal Consistent Sets: Satisfiability

Theorem 97 (Maximal Consistent Sets Satisfiable)
Every maximal consistent set is satisfiable (i.e., has a model).

Proof. Suppose Γ is a maximal consistent set in $\mathscr{L}_{\to,\neg}$. For s a sentence letter, define a structure $\mathscr{A}$: $\mathscr{A}(s) = T$ iff $s \in \Gamma$.

For any ϕ, $[\![\phi]\!]_{\mathscr{A}} = T$ iff $\phi \in \Gamma$. Proof by induction on complexity of sentences. The base case, ϕ a sentence letter, is immediate.

Suppose ϕ is complex. There are two cases:

1. $\phi = \neg\psi$. $\phi \in \Gamma$ iff $\psi \notin \Gamma$, by Negation Completeness; iff by the induction hypothesis, $[\![\psi]\!]_{\mathscr{A}} = F$; iff by the clause on $\neg$, $[\![\phi]\!]_{\mathscr{A}} = T$.

2. $\phi = (\psi \to \chi)$. $\phi \in \Gamma$ iff if $\psi \in \Gamma$ then $\chi \in \Gamma$ (by Conditional Completeness); iff if $[\![\psi]\!]_{\mathscr{A}} = T$ then $[\![\chi]\!]_{\mathscr{A}} = T$; iff $[\![\phi]\!]_{\mathscr{A}} = T$. □

Completeness

Theorem 98 (Completeness for $\mathscr{L}_{\neg,\to}$)
If $\Gamma \vDash \phi$, then $\Gamma \vdash \phi$.

Proof. We prove the *contrapositive*. So assume that $\Gamma \nvdash \phi$. (Assuming that Γ itself is consistent; if it is not, then clearly the theorem holds fairly trivially.)

We'll construct a maximal consistent set Γ^+, a *superset* of Γ, such that $\Gamma^+ \nvdash \phi$. By consistency of Γ^+, $\phi \notin \Gamma^+$; by the Negation Completeness lemma, $\neg\phi \in \Gamma^+$.

But by the Satisfiability Theorem, Γ^+ has a model, $\mathscr{A}$. Since $\Gamma \subseteq \Gamma^+$, for all $\gamma \in \Gamma$, $[\![\gamma]\!]_{\mathscr{A}} = T$; but since $\neg\phi \in \Gamma^+$, $[\![\neg\phi]\!]_{\mathscr{A}} = T$, so $[\![\phi]\!]_{\mathscr{A}} = F$. So $\Gamma \nvDash \phi$, as required.

We now show how to construct Γ^+ from Γ, completing the proof. □

The Construction of Maximal Consistent Γ^+ Supposing $\Gamma \nvdash \phi$, construct Γ^+ as follows.

1. Enumerate the sentences of $\mathscr{L}_{\neg,\to}$, $\{\sigma_1, \sigma_2, \dots\}$.

2. Let $\Gamma_0 = \Gamma$.

3. $$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\neg\sigma_{n+1}\} & \text{if } \Gamma_n, \sigma_{n+1} \vdash \phi; \\ \Gamma_n \cup \{\sigma_{n+1}\} & \text{otherwise (i.e., if } \Gamma_n, \sigma_{n+1} \nvdash \phi). \end{cases}$$

4. Let $\Gamma^+ = \bigcup_i \Gamma_i$.

It is clear that $\Gamma \subseteq \Gamma^+$. It is also clear that $\Gamma^+ \nvdash \phi$, for at no stage was a sentence added to any Γ_n that permitted a proof of ϕ.

Finally, Γ^+ is a *maximal consistent set*, since (by construction), for every σ_i, either $\sigma_i \in \Gamma^+$ or $\neg\sigma_i \in \Gamma^+$. Thus Γ^+ has the properties required to prove Completeness stated just above.

Completeness for $\mathscr{L}_1$ We've shown the Completeness Theorem for the restricted language $\mathscr{L}_{\neg,\to}$. But by Expressive Adequacy, any sentence of $\mathscr{L}_1$ can be expressed in $\mathscr{L}_{\neg,\to}$. So we can assure ourselves that any truth function expressible in $\mathscr{L}_1$ can be expressed in $\mathscr{L}_{\neg,\to}$, and any valid argument in $\mathscr{L}_1$ has a corresponding provably correct sequent in $\mathscr{L}_{\neg,\to}$.

For completeness of $\mathscr{L}_1$, we cannot rest there. What if the rules of $\mathscr{L}_1$ don't suffice to show the equivalence of some arbitrary $\mathscr{L}_1$ sentence ϕ with a sentence

only involving $\neg$ and $\rightarrow$? In that case, while a sentence logically equivalent to ϕ is provable, ϕ may not be.

We thus need additional lemmas showing that maximal consistent sets

- Contain $\phi \wedge \psi$ iff they contain both ϕ and ψ;
- Contain $\phi \vee \psi$ iff they contain at least one of ϕ or ψ;
- Contain $\phi \leftrightarrow \psi$ iff they contain ϕ iff they contain ψ.

Then the Satisfiability Theorem must be extended to include these new cases; and the remainder of the theorem is proved as above.

Compactness Again Consideration of the nature of natural deduction proofs should convince you that any conclusion ϕ drawn on the basis of assumptions Γ can be constructed by some finite application of the rules to finitely many sentences in Γ. Provability is essentially a finite notion: proofs are things that can be *constructed*.

This observation, informal though it is, *can* be made precise, to yield another proof of the *Compactness theorem*. Since, by Completeness, every correct semantic sequent $\Gamma \vDash$ is provable, and every proof is a finite object using only finitely many members of Γ, then by Soundness there is a correct semantic sequent $\Gamma' \vDash$ where Γ' is a finite subset of Γ.

9.3 Axioms

Axiomatic Systems Another proof system we will now briefly consider is an *axiomatic* (or Hilbert-style) proof system.

This is familiar in mathematics: one begins with some *axioms* (normally interpreted as 'obvious' or 'certain' truths), and by applications of some very simple truth-preserving rules of inference, one derives further truths.

If one's axioms are well chosen, one should be able to prove all and only truths about a given subject matter from the axioms characterising that subject matter. This was Euclid's method in the *Elements*: he showed that all and only the truths of Euclidean geometry could be established on the basis of his well chosen axioms.

Axioms for Sentential Logic In an axiomatic system, one begins with theorems, the axioms, and applies the rules of inference to preserve theoremhood.

Our axiom system Ł is very simple. It has three *axiom-schemata*, in the language $\mathscr{L}_{\neg,\to}$:

A1 $\vdash_{Ł} (\phi \to (\psi \to \phi))$;

A2 $\vdash_{Ł} ((\phi \to (\psi \to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi)))$;

A3 $\vdash_{Ł} ((\neg\psi \to \neg\phi) \to (\phi \to \psi))$.

Any sentence of $\mathscr{L}_{\neg,\to}$ can be substituted into these axiom schemata to generate an instance of an axiom. Any axiom is a theorem.

The system has one rule (apart from substitution): Modus Ponens, then rule that if $\vdash_{Ł} \phi$ and $\vdash_{Ł} \phi \to \psi$, then $\vdash_{Ł} \psi$.

A *proof* in Ł is any sequence of theorems, each of which is either an axiom or follows by *modus ponens* from earlier theorems.

A Proof in Ł The system Ł is elegant, but proofs in it can be unnatural to say the least: (I annotate lines of the proof with the axiom scheme used.)

1. $\vdash_{Ł} (P \to ((P \to P) \to P)) \to ((P \to (P \to P)) \to (P \to P))$ (A2)
2. $\vdash_{Ł} (P \to ((P \to P) \to P))$ (A1)
3. $\vdash_{Ł} ((P \to (P \to P)) \to (P \to P))$ (MP 1, 2)
4. $\vdash_{Ł} (P \to (P \to P))$ (A1)
5. $\vdash_{Ł} (P \to P)$ (MP 3, 4)

This is (apparently) the *shortest* proof of $P \to P$ in Ł!

Soundness of Ł It is a trivial exercise (and hence left for an exercise) to show that Ł is sound. That is, show:

- All the axioms of Ł are tautologies.
- The rule of Ł preserves tautologousness.

Equivalence of Ł and $ND_{\neg,\rightarrow}$ We now show that, perhaps surprisingly, the proof systems Ł and $ND_{\neg,\rightarrow}$ are equivalent – everything that can be proved in one can be proved in the other. This shows, too, that Ł is sound and complete.

Everything provable in Ł is provable in ND

Theorem 99
If $\vdash_Ł \phi$ *then* $\vdash \phi$.

Proof. The proof is straightforward: show that each axiom is provable in ND, and show that the rule *modus ponens* is a valid rule in ND.

The second part is easy; if $\vdash \phi$ and $\vdash \phi \rightarrow \psi$, then appending an instance of $\rightarrow$Elim to the preceding proofs is a proof of ψ.

Then it is just a matter of proving the axioms. I show A1:

$$\dfrac{\dfrac{[\phi]}{\psi \rightarrow \phi}\rightarrow \text{Intro}}{\phi \rightarrow (\psi \rightarrow \phi)}\rightarrow \text{Intro}$$

□

Proofs from Assumptions in Ł The ugly proof of $\vdash_Ł P \rightarrow P$ two paragraphs back shows the benefits of working with *assumptions*. A proof of ϕ from assumptions Γ in Ł is a sequence of axioms, *members of* Γ, or follows from preceding members by *modus ponens*, and the last line of which is ϕ. This is a legitimate notion, because one can show:

Theorem 100 (Deduction Theorem for Ł)
$\Gamma \vdash_Ł \phi$ *iff for some finite set of sentences* $\gamma_1, \ldots, \gamma_n$ *actually used in the proof of* ϕ, $\vdash_Ł (\gamma_1 \rightarrow (\gamma_2 \rightarrow \ldots (\gamma_n \rightarrow \phi)\ldots))$.

It suffices to show that $\Gamma, \psi \vdash_Ł \phi$ iff $\Gamma \vdash_Ł \psi \rightarrow \phi$. I leave the proof for an exercise.

Everything Provable in ND is provable in Ł

Theorem 101
If $\vdash \phi$, *then* $\vdash_Ł \phi$.

Proof. Induction on length of proofs: we show that the shortest proofs of ND are provable in Ł, and that the rules are provable transitions.

Certainly the shortest proof, the single node ϕ, shows that $\phi \vdash \phi$; but $\phi \models_{Ł} \phi$ is shown by the deduction theorem given $\models_{Ł} \phi \to \phi$.

Suppose we extend some existing proofs by applying the rules of ND:

→Elim We have $\Gamma \vdash \phi$ and $\Gamma \vdash \phi \to \psi$, and extend by →Elim. But this is obviously *modus ponens*, so $\Gamma \models_{Ł} \psi$.

The case of →Intro is shown by the deduction theorem. I leave you to show the cases for negation. □

Further Reading

Natural deduction was invented by Gentzen (1969), the founder of proof theory. The proof theory of natural deduction systems discussed in section 4 of this chapter is based on Prawitz (2006: 39–41); he uses Theorems 91 and 92, and the last problem, and proves the famous 'cut elimination' theorem for natural deduction, unfortunately a topic beyond the scope of these notes.

The axiom system here was developed by Jan Łukasiewicz, simplifying Frege's system: see Borkowski and Słupecki (1958: p. 25). (The Polish notation used in this article is initially confusing, but has the wonderful feature that it is entirely unambiguous without the use of parentheses.) See also Bostock (1997: ch. 5–6); he gives a direct proof of the completeness of a system based on Ł at pp. 217–9.

Exercises

1. Using the model soundness proof for the restricted natural deduction system $ND_{\neg,\to}$, show, by analysing proofs constructed using the remaining rules of the full Halbach system ND, that the system ND is sound.
2. The Completeness proof for $\mathscr{L}_{\neg,\to}$ relied on various lemmas concerning the behaviour of maximal consistent sets with regard to the connectives ¬ and →. Show that
   (a) analogous lemmas hold for ∧, ∨ and ↔.
   (b) the Satisfiability Theorem holds for the full language $\mathscr{L}_1$.
   (c) the full Natural Deduction system is complete for $\mathscr{L}_1$ with respect to the intended semantics.

3. (a) Show that the axiomatic system Ł introduced is sound.

   (b) Show that the axioms A2 and A3 of Ł are provable in natural deduction:

      i. A2. $\vdash_{Ł} ((\phi \to (\psi \to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi)))$;

      ii. A3. $\vdash_{Ł} ((\neg\psi \to \neg\phi) \to (\phi \to \psi))$.

4. Suppose that $\Gamma, \psi \vdash_{Ł} \phi$, shown by a proof Π. Define the *ψ-transform* of Π, written Π^{ψ}, as the sequence that results from prefixing '$\psi \to$' to the front of every sentence in Π. Show that each sentence in Π^{ψ} is provable from assumptions Γ in Ł. Since obviously ψ appears on Π, $\psi \to \psi$ appears on Π^{ψ}, and that is provable from Γ. So you need to show that

   (a) the prefixed axioms (i.e., those $\psi \to \delta$ in Π^{ψ} where δ is an axiom) are provable from Γ (hint: use A1);

   (b) the prefixed assumptions (i.e., those $\psi \to \gamma$ in Π^{ψ} where $\gamma \in \Gamma$) are provable from Γ;

   (c) If χ followed by *modus ponens* from early claims in Π, $\psi \to \chi$ follows from earlier prefixed claims in Π^{ψ} (hint: use A2).

   Using these results, show the deduction theorem for Ł.

5. Show the other cases for the equivalence of *ND* and Ł:[1]

   (a) Show that if $\Gamma, \phi \vdash_{Ł} \psi$ and $\Gamma, \phi \vdash_{Ł} \neg\psi$, then $\Gamma \vdash_{Ł} \neg\phi$.

   (b) Show that if $\Gamma, \neg\phi \vdash_{Ł} \psi$ and $\Gamma, \neg\phi \vdash_{Ł} \neg\psi$, then $\Gamma \vdash_{Ł} \phi$. (Equivalently, show that if $\Gamma \vdash_{Ł} \neg\neg\phi$, then $\Gamma \vdash_{Ł} \phi$.)

[1] These proofs are quite elusive; I propose treating this question as optional.

Part IV

Predicate Logic

Chapter 10

The Syntax and Semantics of $\mathscr{L}_2$

10.1 Syntax of $\mathscr{L}_2$

$\mathscr{L}_2$ We now have a good grip on $\mathscr{L}_1$. But many valid arguments in English *can't* be captured by a valid argument in sentential logic.

One reason for this is the lack of *non-truth-functional* connectives in $\mathscr{L}_1$ (consider English 'possibly', 'it will be that', 'it ought to be that', or even 'S knows that'). We are not going to remedy this lack now.

Another reason for the inability of $\mathscr{L}_1$ to capture valid arguments is that the analysis $\mathscr{L}_1$ provides of an English sentence stops at the truth-functional structure of the sentence. But English sentences have additional *internal structure* which arguments may make use of.

Consider: 'John loves James; therefore, John loves someone'. This is valid, but the best $\mathscr{L}_1$ formalisation is the incorrect sequent: $P \vDash Q$. The validity of this argument depends on the internal structure of the sentences. So now we look at a logic with the resources to formalise this additional structure.

The Syntax of $\mathscr{L}_2$: $\mathscr{L}_1$ alphabet and Predicate Letters The alphabet of $\mathscr{L}_2$– the constituents out of which $\mathscr{L}_2$-sentences may be formed – includes the alphabet

of $\mathscr{L}_1$. So it contains infinitely many sentence letters, as well as the logical connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$, and the parentheses (,).

The alphabet of $\mathscr{L}_2$ contains *predicate letters*. A predicate letter has *argument places* (intuitively, representing the number of designators or variables required to make a grammatical formula). The number of argument places of a predicate letter is called its *arity*. For each arity $n \geqslant 0$, we introduce infinitely many predicate letters: $P^n, Q^n, R^n, P^n_1, Q^n_1, R^n_1, \ldots$. Where $n = 0$, we omit the index. (So any expression of the form P^n_k, Q^n_k, or R^n_k, where n, k are either missing or a numeral '1', '2', etc. will be a predicate letter.)

We permit zero-place predicate letters, and identify these with the sentence letters of $\mathscr{L}_1$.

The Syntax of $\mathscr{L}_2$: Constants, Variables, Quantifiers The alphabet of $\mathscr{L}_2$ includes infinitely many *constants*, $a, b, c, a_1, b_1, c_1, \ldots$.

$\mathscr{L}_2$ also contains infinitely many *variables*, $x, y, z, x_1, y_1, z_1, \ldots$.

And $\mathscr{L}_2$ contains infinitely many *quantifiers*, $\forall\upsilon$ and $\exists\upsilon$, where υ is any variable. $\forall$ is the *universal* quantifier; $\exists$ is the *existential* (sometimes, *particular*) quantifier.

Variables and constants are collectively known as *terms*.

Atomic Formulae and Formulae of $\mathscr{L}_2$

Definition 104 (Atomic Formula). If Φ^n is a predicate letter of arity $n \geqslant 0$, and $\tau_1, \ldots, \tau_n$ are n terms (not necessarily all distinct), then $\Phi\tau_1 \ldots \tau_n$ is an *atomic formula* of $\mathscr{L}_2$.

It follows that all sentence letters are atomic formulae.

Definition 105 (Formula).

1. Any atomic formula is a formula of $\mathscr{L}_2$.
2. If ϕ and ψ are formulae of $\mathscr{L}_2$, then so are $\neg\phi$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$.
3. If $\forall\upsilon$ and $\exists\upsilon$ are quantifiers and ϕ is a formula, then so are $\forall\upsilon\phi$ and $\exists\upsilon\phi$.

4. Nothing else is a formula of $\mathscr{L}_2$.

We permit brackets and the arity of predicate letters to be dropped in accordance with Halbach, §4.4.

Free and Bound Occurrences of Variables

Definition 106 (Free and Bound Variable Occurrences).

1. All variable occurrences in atomic formulae are free.
2. If some variables occur freely in ϕ and ψ, they remain free in $\neg\phi$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$.
3. No occurrence of v is free in $\forall v\phi$ or $\exists v\phi$; all other variable occurrences in ϕ remain free in $\forall v\phi$.

A variable occurrence is *bound* iff it is not free.

A variable *occurs freely* in ϕ iff there is at least one free occurrence of that variable in ϕ. If a variable occurs freely in ϕ, ϕ is an *open formula.*

Definition 107 (Sentence of $\mathscr{L}_2$). An $\mathscr{L}_2$-formula ϕ is a *sentence* (or 'closed formula') of $\mathscr{L}_2$ iff no variable occurs freely in ϕ.

10.2 Semantics of $\mathscr{L}_2$

Semantics for $\mathscr{L}_2$: Domains As in $\mathscr{L}_1$, the idea is to specify meanings (or *extensions*) for the basic parts of the language, and systematically show how the meaning of complex expressions depends on the meanings of the parts and their relations. But now we have sub-sentential constituents of formulae, which cannot be assigned truth-values as their extensions, so $\mathscr{L}_1$ structures cannot suffice for $\mathscr{L}_2$.

We begin with the notion of a *domain D*. This is a non-empty set of objects – any objects – from which are drawn the *semantic values* of constants. (So the meaning of a constant will be an object.) The meaning of a predicate letter will be a *property* or *relation* on the domain. For our purposes, a subset of the domain

will suffice for a property (intuitively, those things in the domain which have the property); and a subset of $D \times D$ will suffice for a 2-place relation. In general, then, the semantic value of a n-ary predicate letter will be a subset of $D \underbrace{\times \ldots \times}_{n} D = D^n$.

The attentive reader will have noticed a potential oddity here. The semantic value assigned to a 2-place predicate letter will be a set of pairs, and that assigned to a 3-place predicate letter will be a set of triples, etc. Shouldn't the semantic value assigned to a 1-place monadic predicate letter be a set of 1-tuples, rather than a subset of the domain? That is, shouldn't the semantic value of a predicate be something like $X = \{\langle x_1 \rangle, \ldots, \langle x_n \rangle, \ldots\}$, rather than $X' = \{x_1, \ldots, x_n, \ldots\}$? The still more attentive reader will recall that, by Definition 15, $\langle x_i \rangle = x_i$, so that $X = X'$. A subset of the domain turns out to be a set of ordered 1-tuples, so the interpretation of all n-ary predicate letters is uniform: a set of n-tuples.

$\mathscr{L}_2$ Structures and Variable Assignments

Definition 108 ($\mathscr{L}_2$-Structure). An $\mathscr{L}_2$-structure $\mathscr{A}$ is an ordered pair $\langle D, I \rangle$ where D ('the domain') is any *non-empty* set, and I ('the interpretation') is a function

- which assigns an element of D to every constant;
- which assigns a subset of D^n to every predicate letter of arity $n > 0$;
- and which assigns a truth value (T or F) to any predicate letter of arity 0 (sentence letter).

Variables do not have a fixed extension, even in a particular structure. But they may be 'temporarily' assigned semantic values:

Definition 109 (Variable Assignment). A *variable assignment* α over $\mathscr{A}$ is a function which assigns each variable in $\mathscr{L}_2$ a member of $D_{\mathscr{A}}$.

Semantic Values of Simple Expressions Let us extend the notation $[\![\cdot]\!]_{\mathscr{A}}$ to mean the semantic value of any $\mathscr{L}_2$ expression in structure $\mathscr{A}$.

For formulae not involving variables, the assignment of semantic values (in particular, truth values) to complex formulae in a given structure will be straightforward. But what to do about variables – in particular, assigning semantic values

to open formulae (those with free variables) and quantified formulae – poses some delicate issues. Here I use an approach due to Tarski; later I explore some alternatives.

Begin by defining, for structure $\mathscr{A}$ and variable assignment α, $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = I_{\mathscr{A}}(\phi)$ for all ϕ in the domain of $I_{\mathscr{A}}$. (All constants, and predicate letters, including sentence letters, thus receive a semantic value at this stage.) The variables of $\mathscr{L}_2$ get the obvious interpretation: $[\![\upsilon]\!]^{\alpha}_{\mathscr{A}} = \alpha(\upsilon)$. We now assign truth values to the formulae of $\mathscr{L}_2$.

Satisfaction

Definition 110 (Satisfaction). Supposing ϕ, ψ to be formulae of $\mathscr{L}_2$, υ a variable, $\mathscr{A}$ a $\mathscr{L}_2$-structure, and α a variable assignment over $\mathscr{A}$. We say that α *satisfies* ϕ under $\mathscr{A}$, $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = T$, as follows:

- $[\![\Phi\tau_1 \dots \tau_n]\!]^{\alpha}_{\mathscr{A}} = T$ iff $\langle [\![\tau_1]\!]^{\alpha}_{\mathscr{A}}, \dots, [\![\tau_n]\!]^{\alpha}_{\mathscr{A}} \rangle \in [\![\Phi]\!]^{\alpha}_{\mathscr{A}}$, where Φ is a n-ary predicate letter ($n > 0$), and each τ_i is a term.[1]
- $[\![\neg\phi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = F$.
- $[\![\phi \wedge \psi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\psi]\!]^{\alpha}_{\mathscr{A}} = T$.
- $[\![\phi \vee \psi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = T$ or $[\![\psi]\!]^{\alpha}_{\mathscr{A}} = T$.
- $[\![\phi \rightarrow \psi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = F$ or $[\![\psi]\!]^{\alpha}_{\mathscr{A}} = T$.
- $[\![\phi \leftrightarrow \psi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\psi]\!]^{\alpha}_{\mathscr{A}}$.
- $[\![\forall\upsilon\phi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\beta}_{\mathscr{A}} = T$ for all variable assignments β over $\mathscr{A}$ differing from α *at most* in their assignment to υ.
- $[\![\exists\upsilon\phi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\beta}_{\mathscr{A}} = T$ for at least one variable assignment β over $\mathscr{A}$ differing from α *at most* in its assignment to υ.

[1] Recall the discussion of ordered 1-tuples at the beginning of this section when applying this definition to monadic predicates.

Explaining Satisfaction The definition of satisfaction may appear puzzling at first.

The clause for atomic formulae is fairly straightforward. A variable assignment satisfies an atomic formula in a structure just in case the ordered sequence of the objects which are the semantic values of the constants and variables, under that assignment, have the property or stand in the relation which is the semantic value of the predicate letter.

The quantifier clauses are also fairly clear:

- A universally quantified sentence $\forall \upsilon \phi$ is satisfied just in case ϕ is satisfied in every variable assignment that holds everything fixed except possibly the semantic value of υ – i.e., no matter what υ denotes, ϕ is satisfied.
- An existentially quantified sentence $\exists \upsilon \phi$ is satisfied just in case ϕ is satisfied in at least one variable assignment that holds everything fixed except possibly the semantic value of υ – i.e., there is a semantic value for υ on which ϕ is satisfied.

Truth The notion of truth is then defined in terms of satisfaction:

Definition 111 (Truth). A sentence ϕ is *true* in $\mathscr{A}$ iff $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = T$ for every variable assignment α over $\mathscr{A}$.

Notice that while open formulae can be always satisfiable, they cannot be true – that property is reserved for sentences. (If we had not restricted the notion, the open formula $Px \vee \neg Px$, which is assigned true under every variable assignment, would count as true – yet intuitively, at most it expresses a truth conditional on some constant being substituted for x, not unconditionally as a sentence does.)

10.3 Some Semantic Theorems About $\mathscr{L}_2$

Semantic Sequents for $\mathscr{L}_2$ We now extend the semantic turnstile $\vDash$ to $\mathscr{L}_2$. With the new definition of satisfaction and truth in a model in hand, we can define the old notions in much the same way:

1. If ϕ is true in all $\mathscr{L}_2$-structures, ϕ is a *tautology* or *logical* truth, written $\vDash \phi$.

2. If ϕ is true in no $\mathscr{L}_2$-structure, ϕ is a *contradiction*, written $\phi \vDash$.

3. A set Γ of $\mathscr{L}_2$ sentences is *semantically consistent* iff there is some $\mathscr{L}_2$-structure in which each $\gamma \in \Gamma$ is true; otherwise Γ is inconsistent, written $\Gamma \vDash$.

4. ϕ is *logically equivalent* to ψ iff both sentences are true in the same $\mathscr{L}_2$-structures, written $\phi \equiv \psi$. (Some logicians use this symbol to represent the object language biconditional connective, which we represent by $\leftrightarrow$ – do not be confused.)

Definition 112 (Validity in $\mathscr{L}_2$). An argument from Γ to ϕ is *valid* (Γ entails ϕ) iff in every $\mathscr{L}_2$-structure in which each $\gamma \in \Gamma$ is true, ϕ is true also. We write again $\Gamma \vDash \phi$.

Some Theorems About Entailment in $\mathscr{L}_2$ Many of the theorems about $\vDash$ we proved for $\mathscr{L}_1$ in Chapter 2 still hold, with exactly the same proofs.

Structural Rules All the structural rules (Permutation, Weakening, and Contraction) still hold.

Cut If $\Gamma, \phi \vDash \psi$ and $\Gamma \vDash \phi$, then $\Gamma \vDash \psi$.

Transitivity If $\Gamma \vDash \phi$ and $\phi \vDash \psi$, then $\Gamma \vDash \psi$.

Contraposition If $\phi \vDash \psi$ iff $\neg\psi \vDash \neg\phi$.

Deduction Theorem $\Gamma, \phi \vDash \psi$ iff $\Gamma \vDash \phi \to \psi$.

We now prove some theorems that depend on the new machinery we've introduced.

Quantifier Interdefinability

Theorem 102 (Interdefinable Quantifiers)
For all structures $\mathscr{A}$, and all variable assignments α,

- $[\![\forall v\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\neg\exists v\neg\phi]\!]^{\alpha}_{\mathscr{A}}$;
- $[\![\exists v\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\neg\forall v\neg\phi]\!]^{\alpha}_{\mathscr{A}}$.

Proof. I show only the first case. $[\![\forall \upsilon \phi]\!]^{\alpha}_{\mathscr{A}} = T$ iff for *all* υ-variant assignments β, $[\![\phi]\!]^{\beta}_{\mathscr{A}} = T$; iff $[\![\neg\phi]\!]^{\beta}_{\mathscr{A}} = F$; iff $[\![\exists \upsilon \neg \phi]\!]^{\alpha}_{\mathscr{A}} = F$ (if not, there would be a β where $[\![\neg\phi]\!]^{\beta}_{\mathscr{A}} = T$); iff $[\![\neg\exists \upsilon \neg \phi]\!]^{\alpha}_{\mathscr{A}} = T$. □

Substitution of Co-Designating Terms

Theorem 103 (Substitution of Co-Designators)
Let τ_1 and τ_2 be terms, and let $\phi[\tau_2/\tau_1]$ be the formula which results from substituting τ_2 for at least some free occurrences of τ_1 in ϕ, where none of those occurrences is in the scope of $\forall\tau_2$ or $\exists\tau_2$ (let constants vacuously occur free). Then, for any structure $\mathscr{A}$ and any variable assignment α: If $[\![\tau_1]\!]^{\alpha}_{\mathscr{A}} = [\![\tau_2]\!]^{\alpha}_{\mathscr{A}}$, then $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\phi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}}$.

Proof. By induction on length of formulae.

Base case: Suppose ϕ is an atomic formula, $\Phi\theta_1 \dots \theta_n$, where perhaps some $\theta_i = \tau_1$. But $[\![\Phi\theta_1 \dots \theta_n]\!]^{\alpha}_{\mathscr{A}} = T$ iff $\langle [\![\theta_1]\!]^{\alpha}_{\mathscr{A}}, \dots, [\![\tau_1]\!]^{\alpha}_{\mathscr{A}}, \dots, [\![\theta_n]\!]^{\alpha}_{\mathscr{A}}\rangle \in [\![\Phi]\!]^{\alpha}_{\mathscr{A}}$; since $[\![\tau_1]\!]^{\alpha}_{\mathscr{A}} = [\![\tau_2]\!]^{\alpha}_{\mathscr{A}}$, $\langle [\![\theta_1]\!]^{\alpha}_{\mathscr{A}}, \dots, [\![\tau_2]\!]^{\alpha}_{\mathscr{A}}, \dots, [\![\theta_n]\!]^{\alpha}_{\mathscr{A}}\rangle \in [\![\Phi]\!]^{\alpha}_{\mathscr{A}}$, i.e., $[\![\phi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$.

Induction step: Suppose ϕ is complex, and the theorem holds for ψ, χ less complex than ϕ and *each* variable assignment α over $\mathscr{A}$. We consider some representative cases, leaving others for exercises:

- For the first case, suppose $\phi = \neg\psi$. It follows from the induction hypothesis that $[\![\neg\psi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\neg\psi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$; by the clause on negation, that holds iff $[\![(\neg\psi)[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$, i.e., iff $[\![\phi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$.

- Suppose $\phi = \psi \vee \chi$. $[\![\psi \vee \chi]\!]^{\alpha}_{\mathscr{A}} = T$ iff either $[\![\psi]\!]^{\alpha}_{\mathscr{A}} = T$ or $[\![\chi]\!]^{\alpha}_{\mathscr{A}} = T$; by the induction hypothesis, iff $[\![\psi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$ or $[\![\chi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$; by substitution, iff $[\![(\psi \vee \chi)[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$, i.e., iff $[\![\phi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$.

- Suppose $\phi = \forall \upsilon \psi$. $[\![\forall \upsilon \psi]\!]^{\alpha}_{\mathscr{A}} = T$ iff for any variable assignment β which agrees with α except perhaps on υ, $[\![\psi]\!]^{\beta}_{\mathscr{A}} = T$. By the induction hypothesis, and the fact that it holds for all variable assignments, that holds iff $[\![\psi[\tau_2/\tau_1]]\!]^{\beta}_{\mathscr{A}} = T$ (note here we use the assumption that τ_1 doesn't occur in the scope of any quantifier that would bind τ_2). That holds iff $[\![\forall \upsilon \psi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$ (again, it is required $\upsilon \neq \tau_2$), i.e., iff $[\![\phi[\tau_2/\tau_1]]\!]^{\alpha}_{\mathscr{A}} = T$. □

Satisfaction of Sentences

Theorem 104 (Satisfaction of Sentences)
If ϕ is a $\mathscr{L}_2$-sentence, then for every structure $\mathscr{A}$ and all variable assignments α, β, $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\phi]\!]^{\beta}_{\mathscr{A}}$.

Proof. by induction on complexity. *Base case*: ϕ is an atomic formula. It is a sentence, so no variables occur in $\phi = \Phi\tau_1 \ldots \tau_n$; since constants receive a constant interpretation in A, if ϕ is satisfied by any variable assignment it is satisfied by any other.

Induction step: Suppose the theorem holds for less complex sentences ψ and all variable assignments over $\mathscr{A}$. The only interesting cases are the quantifiers, so let $\phi = \forall \upsilon \chi$.

1. χ is a sentence (so the quantification is vacuous). Then for any α, β $[\![\chi]\!]^{\alpha}_{\mathscr{A}} = [\![\chi]\!]^{\beta}_{\mathscr{A}}$, so $[\![\forall \upsilon \chi]\!]^{\alpha}_{\mathscr{A}} = [\![\forall \upsilon \chi]\!]^{\beta}_{\mathscr{A}}$.

2. χ is not a sentence; since ϕ is, only υ occurs free in χ. Suppose for some α, β, $[\![\phi]\!]^{\alpha}_{\mathscr{A}} \neq [\![\phi]\!]^{\beta}_{\mathscr{A}}$. Without loss of generality, suppose $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = \boldsymbol{T}$. Then for every variable assignment γ differing from α at most on υ, $[\![\chi]\!]^{\gamma}_{\mathscr{A}} = \boldsymbol{T}$.

   But for at least one variable assignment δ differing from β on at most υ, $[\![\chi]\!]^{\delta}_{\mathscr{A}} = \boldsymbol{F}$.

   Let τ not occur in χ, and let $[\![\tau]\!]^{\delta}_{\mathscr{A}} = [\![\upsilon]\!]^{\delta}_{\mathscr{A}}$. Then by the Substitution of Terms Theorem, $[\![\chi]\!]^{\delta}_{\mathscr{A}} = [\![\chi[\tau/\upsilon]]\!]^{\delta}_{\mathscr{A}} = \boldsymbol{F}$. By the induction hypothesis, for *every* γ, $[\![\chi[\tau/\upsilon]]\!]^{\gamma}_{\mathscr{A}} = \boldsymbol{F}$. In particular, for some γ^* which is like α except that it assigns $[\![\tau]\!]_{\mathscr{A}}$ to be the extension of υ, $[\![\chi[\tau/\upsilon]]\!]^{\gamma^*}_{\mathscr{A}} = \boldsymbol{F}$. But since $[\![\tau]\!]^{\gamma^*}_{\mathscr{A}} = [\![\upsilon]\!]^{\gamma^*}_{\mathscr{A}}$, $[\![\chi]\!]^{\gamma^*}_{\mathscr{A}} = \boldsymbol{F}$; contradicting our assumption that every υ-variant of α satisfied χ. So there is no case where $[\![\phi]\!]^{\alpha}_{\mathscr{A}} \neq [\![\phi]\!]^{\beta}_{\mathscr{A}}$. □

Quantifiers: Universal Elimination/Existential Introduction

Theorem 105
Suppose ϕ is a formula in which only υ occurs free, and τ any constant. Then:

- $\forall \upsilon \phi \vDash \phi[\tau/\upsilon]$;
- $\phi[\tau/\upsilon] \vDash \exists \upsilon \phi$.

Proof. I only show the second. Suppose that $\mathscr{A}$ is a structure in which $\phi[\tau/\upsilon]$ is true. Then in every variable assignment α according to which $[\![\tau]\!]^{\alpha}_{\mathscr{A}} = [\![\upsilon]\!]^{\alpha}_{\mathscr{A}}$, $[\![\phi[\tau/\upsilon]]\!]^{\alpha}_{\mathscr{A}} = [\![\phi]\!]^{\alpha}_{\mathscr{A}}$ (by the substitution of co-designating terms). There will be such a variable assignment α; so by the clause for $\exists$, $[\![\exists\upsilon\phi]\!]_{\mathscr{A}} = T$.

Quantifiers: Universal Introduction/Existential Elimination

Theorem 106

Suppose ϕ is a formula in which at most υ occurs free, and τ any constant not occurring in Γ or ϕ. *Then:*

- *If* $\Gamma \vDash \phi[\tau/\upsilon]$, *then* $\Gamma \vDash \forall\upsilon\phi$.
- *If* $\Gamma, \phi[\tau/\upsilon] \vDash \psi$ *then* $\Gamma, \exists\upsilon\phi \vDash \psi$, *provided* τ *does not occur in* ψ.

Proof. Suppose that every structure $\mathscr{A}$ which makes Γ true makes $\phi[\tau/\upsilon]$ true. τ does not occur in Γ, so every structure $\mathscr{A}'$ just like $\mathscr{A}$ except perhaps in the value of $I_{\mathscr{A}'}(\tau)$ also makes all of Γ true. Then $\phi[\tau/\upsilon]$ is true in *every* such τ-variant structure $\mathscr{A}'$. Suppose β is a variable assignment where $[\![\phi]\!]^{\beta}_{\mathscr{A}} = F$. There is some $\mathscr{A}'$ where $I_{A'}(\tau) = [\![\upsilon]\!]^{\beta}_{\mathscr{A}}$, so $[\![\phi[\tau/\upsilon]]\!]_{\mathscr{A}'} = F$. Contradiction; there is no such variable assignment, so $\forall\upsilon\phi$ is true in every structure $\mathscr{A}$ which makes all of Γ true.

Substitution of Equivalent Formulae

Theorem 107 (Substitution of Equivalent Formulae)

Suppose ϕ and ψ are formulae, in which $\upsilon_1, \ldots, \upsilon_n$ occur free. Suppose δ is a some sentence in which ϕ occurs as a subformula. Then $\forall\upsilon_1 \ldots \forall\upsilon_n(\phi \leftrightarrow \psi) \vDash \delta \leftrightarrow \delta[\psi/\phi]$.

Proof. In every structure $\mathscr{A}$ in which $\forall\upsilon_1 \ldots \forall\upsilon_n(\phi \leftrightarrow \psi)$ is true, $\phi \leftrightarrow \psi$ is satisfied by every variable assignment agreeing on variables except possibly the υ_is. But clearly if for every variable assignment α $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\psi]\!]^{\alpha}_{\mathscr{A}}$, a straightforward induction on complexity of δ will establish the result. □

10.4 Alternative Semantics for $\mathscr{L}_2$

Semantics only for sentences *Objection*: '*Why are open formulae being assigned truth values at all? They couldn't express any proposition, and we should only assign truth to formulae which could express propositions: sentences.*'

The easiest way to secure this result is as follows:

- Redefine formulae of $\mathscr{L}_2$ so that no open formula is a formula at all. We will thus alter the formation rules so that we do not add quantifiers to bind variables in open formulae, but rather to (i) *substitute* a variable ξ for zero or more occurrences of some constant τ in ϕ; (ii) say that $\forall\xi\phi[\xi/\tau]$ and $\exists\xi\phi[\xi/\tau]$ are formulae.

- Replace the semantic clauses involving satisfaction with something like this:

  - $[\![\forall\xi\phi]\!]_{\mathscr{A}} = T$ iff for every individual $i \in D_{\mathscr{A}}$, and every structure $\mathscr{A}'$ just like $\mathscr{A}$ except perhaps in the semantic value of τ, if $[\![\tau]\!]_{\mathscr{A}'} = i$, then $[\![\phi[\tau/\xi]]\!]_{\mathscr{A}'} = T$.

  This alternative semantics gives a *recursion on truth* rather than satisfaction.

'Substitutional' Quantification *Objection*: '*I like the idea that universal quantification is kind of like an infinite conjunction of all of its instances. Do we need to make the confusing detour through variable assignments?*'

Here is the idea behind this objection. What $\forall x Px$ really means is that whatever constant a one substitutes for x, Pa will be true. This is made more precise as follows: where ϕ is a formula with at most v free,

- $[\![\forall v\phi]\!]_{\mathscr{A}} = T$ iff for any constant τ, $[\![\phi[\tau/v]]\!]_{\mathscr{A}} = T$.

These alternative truth conditions quickly go awry if there are unnamed objects in the domain. For instance, in an uncountable domain (like the real numbers), as there are only countably many constants in $\mathscr{L}_2$, there are entities which cannot be the semantic value of any constant. Suppose we enumerate the constants a_i, letting $I_{\mathscr{A}}(a_i) = i$. Then on the domain of the real numbers, since every constant

denotes an integer, $\forall x(\text{Integer}(x))$ looks like it will be true in $\mathscr{A}$ on the substitutional reading; yet this is highly counterintuitive.

say something about corner-quotes

Truth for Open Formulae *Objection*: '*Wait! We do have the resources to explain the truth of an open formula: it is true whenever it is true under every variable assignment. So why be so restrictive?*'

On this proposal, $\vDash \phi$ iff ϕ is true in every structure under every variable assignment. In effect, therefore, an open formula ϕ with free variables $v_1, \ldots, v_n$ will be a tautology iff its *universal closure* (the result of binding all free variables with universal quantifiers) $\forall v_1 \ldots \forall v_n \phi$ is a tautology.

Yet this is problematic; for then this will be a correct sequent:

$$Px \vDash \forall x Px.$$

But the deduction theorem then fails! For

$$\vDash Px \to \forall x Px \qquad \text{iff} \qquad \vDash \forall x(Px \to \forall x Px),$$

and the right hand side sequent is clearly incorrect.

Further Reading

Tarski's original definition of satisfaction and truth can be found in Tarski (1933). (This is a translation of his 1936 paper.) Bostock (1997: §§3.4–3.6) gives a semantics, and proves many theorems, in which the basic semantic clauses involve recursion on truth rather than satisfaction. Kripke (1976) contains an interesting though technical discussion of the merits and problems with the substitutional intepretation of the quantifiers.

Exercises

1. Suppose we define a sentence ϕ to be *true** in $\mathscr{A}$ iff $[\![\phi]\!]^\alpha_{\mathscr{A}} = T$ for at least one variable assignment α over $\mathscr{A}$. Show that ϕ is true in $\mathscr{A}$ iff it is true* in $\mathscr{A}$.
2. Prove that, for all structures $\mathscr{A}$ and variable assignments α, $[\![\exists v\phi]\!]^\alpha_{\mathscr{A}} = [\![\neg\forall v\neg\phi]\!]^\alpha_{\mathscr{A}}$.
3. Prove that if v is not free in ϕ, $[\![\forall v\phi]\!]^\alpha_{\mathscr{A}} = [\![\phi]\!]^\alpha_{\mathscr{A}}$. If ϕ is a sentence, show that ϕ and $\forall v\phi$ are logically equivalent.

4. Prove that, if υ and ν are variables, ϕ a formula not containing υ, and $\phi[\upsilon/\nu]$ the formula that results from replacing *every* occurence of ν in ϕ with υ – *even those that occur in quantifiers* $\forall\nu$, $\exists\nu$ – then for any structure $\mathscr{A}$ and any variable assignment α, there exists a variable assignment β on $\mathscr{A}$ such that $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\phi[\upsilon/\nu]]\!]^{\beta}_{\mathscr{A}}$.

5. Prove that, for any structure $\mathscr{A}$, if Φ and Ψ are n-ary predicate letters such that $[\![\Phi]\!]_{\mathscr{A}} = [\![\Psi]\!]_{\mathscr{A}}$, ϕ a formula, and $\phi[\Psi/\Phi]$ the formula that results from replacing *every* occurrence of Φ in ϕ with Ψ, then for any variable assignment α over $\mathscr{A}$, $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = [\![\phi[\Psi/\Phi]]\!]^{\alpha}_{\mathscr{A}}$. Then prove that, if ϕ is a sentence, ϕ is true in $\mathscr{A}$ iff $\phi[\Psi/\Phi]$ is true in $\mathscr{A}$.

6. Suppose ϕ is a formula in which only υ occurs free, and τ any constant. Prove that $\forall\upsilon\phi \vDash \phi[\tau/\upsilon]$.

7. Suppose ϕ is a formula in which at most υ occurs free, and τ any constant *not occurring in* Γ *or* ϕ. Prove that if $\Gamma, \phi[\tau/\upsilon] \vDash \psi$ then $\Gamma, \exists\upsilon\phi \vDash \psi$, provided τ does not occur in ψ.

8. Show that, if τ and θ are any constants:

   (a) If $\Gamma \vDash \phi$ then $\Gamma[\tau/\theta] \vDash \phi[\tau/\theta]$.

   (b) If $\Gamma \vDash$ then $\Gamma[\tau/\theta] \vDash$.

9. Suppose ϕ and ψ are any formulae, in which possibly (but not necessarily) $\upsilon_1, \ldots, \upsilon_n$ occur free. Suppose δ is a some sentence in which ϕ occurs as a subformula. Using induction on complexity of δ, establish the result that $\forall\upsilon_1 \ldots \forall\upsilon_n(\phi \leftrightarrow \psi) \vDash \delta \leftrightarrow \delta[\psi/\phi]$. (Hint: you might want to consider an induction showing joint satisfiability of any formulae γ and $\gamma[\psi/\phi]$.)

10. Show that $\forall\xi(\phi \leftrightarrow \psi) \vDash \forall\xi\phi \leftrightarrow \forall\xi\psi$.

11. Give a full specification of the alternative formation rules for a language $\mathscr{L}_2{}'$ in which there are no open formulae. Show that the sentences of $\mathscr{L}_2$ are the formulae of $\mathscr{L}_2{}'$.

12. Give a semantic clause for $\exists$ on the alternative semantics using *recursion on truth* as specified on page 182.

13. Show the following:

   (a) $\forall x(Px \wedge Qx) \equiv \forall xPx \wedge \forall xQx$;

   (b) $\exists x(Px \wedge Qx) \vDash \exists xPx \wedge \exists xQx$. Why not vice versa?

   (c) $\exists y\forall xPxy \vDash \forall x\exists yPxy$. Why not vice versa?

14. If $v_1 \ldots v_n$ is a sequence of variables, let $\forall v_1 \ldots v_n$ be the string of quantifiers $\forall v_1 \ldots \forall v_n$. Let $\mathscr{P}$ be an arbitrary permutation of the order of a sequence. Show that $\forall v_1 \ldots v_n \phi \equiv \forall \mathscr{P} v_1 \ldots v_n \phi$.

15. Explain why the 'objectual' reading of the quantifiers (i.e., the standard one) has no difficulty with universal quantification over an uncountable domain.

Answers to selected exercises on page 250.

Chapter 11

Tableaux for $\mathscr{L}_2$; Soundness

Chapter 12

Natural Deduction Derivations in $\mathscr{L}_2$; Soundness

12.1 Proofs in $\mathscr{L}_2$

Proofs in $\mathscr{L}_2$ In the case of $\mathscr{L}_1$, a proof procedure was not strictly speaking necessary. Every sentence of $\mathscr{L}_1$ has only finitely many sentence letters, so a truth table will need only to specify finitely many rows to capture what is true in every $\mathscr{L}_1$-structure.

But in $\mathscr{L}_2$ we don't have anything like a truth-table: one cannot simply survey a finite number of cases to represent all the structures. (Intuitively, this is because of the introduction of quantification – now a sentence may depend for its truth on the way infinitely many things are, so surveying only finitely many of them won't suffice.) Brute force checking of semantic sequents is not possible.

Thus the introduction of a proof technique, to enable the derivation of some $\mathscr{L}_2$ claims from others, is crucial. Once more, we use natural deduction; again, we abuse notation at let $\vdash$ stand for *provability in* $\mathscr{L}_2$.

Natural Deduction The rules for the natural deduction system ND_2 include all of those from the original system ND. As $\mathscr{L}_2$ includes all of $\mathscr{L}_1$, everything provable in ND is also provable in ND_2. But we include new rules to cover the quan-

$\dfrac{\forall \upsilon \phi}{\phi[\tau/\upsilon]}\ \forall\text{Elim}$	$\dfrac{\begin{array}{c}\vdots\\ \phi[\tau/\upsilon]\end{array}}{\forall \upsilon \phi}\ \forall\text{Intro}$
Provided τ doesn't occur in ϕ or in any undischarged assumption in the proof of $\phi[\tau/\upsilon]$.	
$\dfrac{\phi[\tau/\upsilon]}{\exists \upsilon \phi}\ \exists\text{Intro}$	$\dfrac{\begin{array}{cc}[\phi[\tau/\upsilon]] & \\ \vdots & \vdots \\ \psi & \exists \upsilon \phi\end{array}}{\psi}\ \exists\text{Elim}$
Provided τ doesn't occur in $\exists \upsilon \phi$, in ψ, or in any undischarged assumption other than $\phi[\tau/\upsilon]$ in the proof of ψ.	

Table 12.1: New Natural Deduction Rules for $\mathscr{L}_2$

tifiers, laid out in Table 12.1. Throughout, let $\phi[\tau/\upsilon]$ be the result of replacing *free* occurrences of variable υ in ϕ by a constant τ.

Some Elementary Theorems About ND_2 Many of the theorems for ND continue to hold. For example, the proof of the *deduction theorem* in ND relied only on the arrow rules, so carry directly over to ND_2.

Theorem 108 (Cut)
If $\Gamma \vdash \phi$ *and* $\phi, \Delta \vdash \psi$ *then* $\Gamma, \Delta \vdash \psi$.

Proof. We cannot, as in ND, simply chain the proofs together, because Γ may contain a constant τ which, in the proof that $\phi, \Delta \vdash \psi$, is used in an application of $\exists$Elim or $\forall$Intro (of course τ cannot appear in ϕ or Δ given the correctness of $\Delta, \phi \vdash \psi$). So we need to show that there is *another* proof of $\Delta, \phi \vdash \psi$ which doesn't involve any such τ. I leave this for an exercise. □

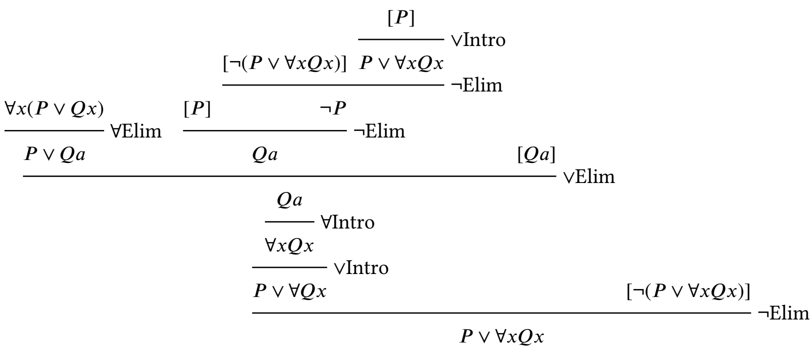
Proofs and Connectives The natural deduction proofs of many sequents involving sentences with given connectives involve only the rules for those connectives. For example, the sequent $\forall x \exists y (Px \wedge Qx) \vdash \exists y \forall x (Px \wedge Qx)$ can be proved entirely using the rules for conjunction and the quantifiers.

But this is not true in general. Consider this: $\forall x (P \vee Qx) \vdash P \vee \forall x Qx$. (Proof below.)

This proof cannot be carried out using just the rules for disjunction and the quantifiers.

The situation is very similar to a case in an earlier problem (Chapter 4, 1.(e), page 159) where the natural deduction rules for → did not suffice to prove every tautology involving → as its only connective.

Proof of $\forall x(P \vee Qx) \vdash P \vee \forall xQx$



Some Proof Theory: Uniqueness of ∃ Some philosophers have wanted to maintain that there are at least two senses of the English word 'exists'. (They have wanted to do this typically to distinguish a committal from a non-committal sense, so they can accept the truth of claims like 'there are at lest two prime numbers less than 8' and 'there are at least two cities smaller than London', while being committed to cities, and avoiding committment to such mysterious entities as numbers.) But these philosophers have typically also desired to speak a language including both quantifiers; or at least to be able to translate between the two languages. There is however a fairly straightforward theorem which shows that this situation is not possible: if there are two quantifiers which both meet the minimal requirements to be an existential quantifier, those quantifiers are logically equivalent.

Theorem 109 (Harris 1982)
If there are two quantifiers $\exists_1$ *and* $\exists_2$ *in a language, both governed by the standard introduction and elimination rules for the quantifiers, then* $\exists_1 \upsilon\phi$ *is logically equivalent to* $\exists_2 \upsilon\phi$*.*

Proof. Assuming that we have introduction and elimination rules for the two

quantifiers, we can give the following natural deduction proof:

$$\dfrac{\exists_1 \upsilon\phi \qquad \dfrac{[\phi[\tau/\upsilon]]}{\exists_2 \upsilon\phi}\,\exists_2\text{Intro}}{\exists_2 \upsilon\phi}\,\exists_1\text{Elim}$$

Obviously a precisely similar proof will show that $\exists_2 \upsilon\phi \vdash \exists_1 \upsilon\phi$. □

Unless, then, we wish to introduce two different sorts of names, two different sorts of predicates, etc., we cannot simply introduce two quantifiers into a language – not, at least, if they are to meet the minimal conditions that quantifiers should, namely, being characterised by our introduction and elimination rules.

12.2 Soundness of $\mathscr{L}_2$

Soundness We are now in possession of a proof system ND_2, which characterises a provability turnstile $\vdash$. We now wish to show that this system is *sound* with respect to the semantics for $\mathscr{L}_2$ we introduced in the last chapter. That is, we wish to prove

Theorem 110 (Soundness)
If $\Gamma \vdash \phi$ *then* $\Gamma \vDash \phi$.

As ND_2 extends ND, and the rules of ND remain sound (verification of this is left to an exercise), we just need to show that the new rules – the quantifier rules – are sound. We also still know that the base case $\phi \vdash \phi$ is sound. So we just consider proofs extended by the quantifier rules.

Soundness of $\mathscr{L}_2$: ∀Elim and ∃Intro Suppose $\Gamma \vdash \phi$, where ϕ is obtained from an earlier proof by the use of the ∀Elim rule, and so is of the form $\psi[\tau/\upsilon]$. Then there was a proof on the assumption Γ of $\forall\upsilon\psi$.

By the induction hypothesis, $\Gamma \vDash \forall\upsilon\psi$. So every structure which makes all the members of Γ true, makes $\forall\upsilon\psi$ true. By theorem 105 (proved on page 180), $\forall\upsilon\psi \vDash \psi[\tau/\upsilon]$. Since $\psi[\tau/\upsilon] = \phi$, ϕ will be true in every structure which makes Γ true, i.e., $\Gamma \vDash \phi$.

The proof for ∃Intro is similar. I leave it for an exercise.

Soundness of $\mathscr{L}_2$: ∀Intro and ∃Elim The trickier case is if $\Gamma \vdash \phi$ which was obtained from ∃Elim. From a proof of $\exists v\psi$ on assumptions Δ, and a proof of ϕ on assumptions $\Theta, \psi[\tau/v]$, we obtain a proof of ϕ on assumption $\Gamma = \Delta \cup \Theta$ (with appropriate restriction on where τ occurs). Since $\Delta \subseteq \Gamma$, we have $\Gamma \vdash \exists v\psi$, and by the induction hypothesis, $\Gamma \vDash \exists v\psi$.

Since $\Theta \subseteq \Gamma$, we have $\Gamma, \psi[\tau/v] \vdash \phi$, and by the induction hypothesis $\Gamma, \psi[\tau/v] \vDash \phi$. By theorem 106 (proved on page 181), $\Gamma, \exists v\psi \vDash \phi$. And by the Cut theorem, $\Gamma \vDash \phi$.

The soundness of ∀Intro is similarly demonstrated; I leave it for an exercise.

12.3 Completeness of $\mathscr{L}_2$

Completeness

Theorem 111 (Completeness of $\mathscr{L}_2$)
If $\Gamma \vDash \phi$, $\Gamma \vdash \phi$.

The proof extends the proof of completeness for $\mathscr{L}_1$, roughly like this:

- First, show that, e.g., if Γ is a maximal consistent set, then $\exists v\phi \in \Gamma$ iff $\phi[\tau/v] \in \Gamma$ for some constant τ. (This is analogous to the other closure properties we showed.)
- Second, show that every maximal consistent set of $\mathscr{L}_2$-sentences is satisfiable.
- Finally, show that if $\Gamma \nvdash \phi$, there is a maximal consistent set Γ^+ such that $\Gamma \cup \{\neg\phi\} \subseteq \Gamma^+$ – since it is maximal, Γ^+ is satisfiable, so there is a model where all of Γ is true and ϕ is false, so $\Gamma \nvDash \phi$. By contraposition, this shows completeness.

We won't prove this theorem here; the details are involved.

Compactness

Theorem 112 (Compactness)
If Γ *is any set of* $\mathscr{L}_2$ *sentences, then* Γ *is satisfiable iff every finite subset of* Γ *is satisfiable.*

Again, compactness is a consequence of the finitude of proofs and the Completeness Theorem. A direct proof, like that we gave for $\mathscr{L}_1$ in Chapter 3, is more difficult in $\mathscr{L}_2$, and again beyond the scope of this course.

Compactness has the consequence that certain relations are undefinable. In English, the infinitely many sentences of the form 'x is *not* a predecessor of…a predecessor of y' entail 'x is not less than y', but no finite subset does – but their translation into the compact language $\mathscr{L}_2$ must therefore fail. This is because the relation between predecessor and less than is not definable in $\mathscr{L}_2$.

12.4 Decidability and Undecidability

Completeness and Decidability

Theorem 113 ($\mathscr{L}_2$ is positively decidable)
If $\vDash \phi$, then there is an effective procedure demonstrating that.

Proof. Completeness shows that if ϕ is a theorem, there is a proof of it. By brute force, we can generate finite proofs and spit out a proof of ϕ if one is to be had. Since the set of sentences of $\mathscr{L}_2$ is countable, and since every proof is of finite length, consider the proofs of length n which involve only the first m $\mathscr{L}_2$ sentences. For each pair $\langle m, n\rangle$ there are only finitely many such proofs; we can effectively produce them. Since the set of pairs of integers are countable (exercise), there is an enumeration of those pairs, hence we can effectively produce any finite proof. But every theorem ϕ is proved by some such finite proof; so at some finite point in the enumeration, we will reach an m greater than the numbers of all the members of a proof of ϕ, and at some finite point spit out a proof of ϕ of length less than n. □

Undecidability

Theorem 114
$\mathscr{L}_2$ *is* negatively undecidable*: if $\phi \nvDash$, no effective procedure exists which will show that for any ϕ in a finite time.*

Considering the effective procedure for positive decidability, we can see why this won't stop after a finite time (the proof of $\neg\phi$ might always be achieved using the

next pair to be considered); but maybe there is another method that will terminate?

There is not. One way of helping to make this thought persuasive is to consider a particular incorrect sequent: $\forall x \exists y Pxy \vDash Paa$. One can certainly give a counterexample to this (let the domain be the natural numbers, and 'P' be interpreted as '$<$'). But is there an effective procedure for producing such counterexamples?

The Halting Problem If S is a finite set of instructions, can we give an effective procedure that determines whether it is the case that those instructions, carried out on some acceptable input, will *terminate* after a finite time with some output?

Consider some enumeration of the set of finite sets of instructions; the halting problem, if soluble, will entail the existence of an effectively computable function $h(i)$ that yields 1 if the i-th set of instructions halts with a defined output on input i, and 0 otherwise (i.e., $h(i) = 0$ if the i-th set of instructions determines a function which is undefined on input i). Can there be such a function h?

Sketch of Insolubility of the Halting Problem If there were such a function h, we could effectively compute the function $g(i) = 0$ if the i-th set of instructions is undefined on input i, and undefined otherwise. We could define g by adding to the instructions that compute h the last instruction: if $h(i)$ outputs 1, then check if $h(i)$ outputs 1. This last instruction will send give an infinite loop if $h(i)$ is equal to 1, rendering $g(i)$ undefined; but if $h(i) = 0$, then $g(i) = 0$ too.

But g, if effectively computable, will be specified by a finite set of instructions; let it be j-th in our list. Then $g(j) = 0$ iff $g(j)$ is undefined, so it is undefined; but then $g(j)$ must be defined. Contradiction; so there is no such g, and therefore no such h. The *self-halting problem* is insoluble, and therefore so is the halting problem.

For a more humorous proof sketch, see Pullum (2008).

Undecidability and the Halting Problem It turns out that the undecidability of $\mathscr{L}_2$ is closely related to the Halting problem. For there is a way of associating to each set of instructions i an *argument* in $\mathscr{L}_2$ such that the i-th set of instructions halts on input i iff the argument is valid. So if there was an effective test for invalidity of an argument in $\mathscr{L}_2$, the halting problem would be solvable. Since it is

insoluble, we have

Theorem 115 (Undecidability of $\mathscr{L}_2$)
$\mathscr{L}_2$ is not decidable.

The translation essentially involves *binary* predicates (Jeffrey §8.7) and sentences in which (sometimes) a universal quantifier is essentially in the scope of an existential quantifier. So we might wonder: are there fragments of $\mathscr{L}_2$ that are decidable?

Prenex Normal Form A sentence ϕ of $\mathscr{L}_2$ is in *prenex normal form* iff no quantifier in ϕ occurs in the scope of any truth-functor (all quantifiers are in a block).

Theorem 116 (PNF)
Every sentence of $\mathscr{L}_2$ is equivalent to a sentence in prenex normal form.

Proof. The proof will be work by showing that if ϕ is some sentence, any quantifier which occurs somewhere in the middle of ϕ can be 'moved' to the left. The idea is to show these equivalences:

1. $\neg\forall v\phi \equiv \exists v\neg\phi$; (basically shown by Theorem 102)
2. $\neg\exists v\phi \equiv \forall v\neg\phi$; (basically shown by Theorem 102)
3. $\psi \wedge \forall v\phi \equiv \forall v(\psi \wedge \phi)$ (v not free in ψ).
4. $\psi \wedge \exists v\phi \equiv \exists v(\psi \wedge \phi)$ (v not free in ψ).

Etc. The full proof is left for an exercise. □

Decidability of ∀∃-sentences An ∀∃-*sentence* is a sentence in PNF in which all universal quantifiers precede all existential quantifiers.

Theorem 117 (Decidability of ∀∃-sentences)
There exists an effective procedure which establishes the validity of any valid ∀∃*-sentence, and the invalidity of any invalid* ∀∃*-sentence.*

The proof is as follows. We show how to reduce the question of validity of a ∀∃-sentence to the question of validity of a related quantifier-free sentence; and such quantifier-free sentences of $\mathscr{L}_2$ are decidable. For convenience, I abbreviate $\forall x_1 \ldots \forall x_n$ as $\forall x_1 \ldots x_n$, and similarly for $\exists$.

Lemma on Quantifier-Free Sentences

Lemma 118 (Quantifier-Free Sentences)

If ϕ is a sentence of $\mathscr{L}_2$, which is not a (substitution instance of a) truth-functional tautology, then there is a structure $\mathscr{A}$ in which $[\![\phi]\!]_{\mathscr{A}} = F$ and either (i) if ϕ contains no constants, the domain of $\mathscr{A}$ has one element; or (ii) otherwise, $\mathscr{A}$ has an element in its domain for each distinct constant occurring in ϕ and no other elements, and each constant in ϕ is assigned a distinct element of the domain.

Proof. Hint: if ϕ is not a truth-functional tautology, there is an assignment of truth-values to atomic sentences in ϕ that mimics an $\mathscr{L}_1$-structure; and we can construct an $\mathscr{L}_2$-structure $\mathscr{A}$ which agrees with that pseudo-$\mathscr{L}_1$-structure on its assignment to atomic sentences. □

A corollary is that quantifier-free sentences of $\mathscr{L}_2$ are decidable, since the truth-table test decides truth-functional tautologies.

∃-sentences

Theorem 119 (Constant-free ∃-sentences)

If ϕ is a quantifier-free formula which contains only the variables $v_1, \dots, v_n$ and which contains no constants, then $\vDash \exists v_1 \dots v_n \phi$ iff $\vDash \phi[\tau/v_1, \dots, \tau/v_n]$.

Proof. R to L: Repeated applications of Existential Introduction (Theorem 105) yield $\phi[\tau/v_1, \dots, \tau/v_n] \vDash \exists v_1 \dots v_n \phi$. By Cut, the result follows.

L to R: If $\nvDash \phi[\tau/v_1, \dots, \tau/v_n]$, then by Lemma 118, there is a structure with a one-element domain $\mathscr{A}$ where it is false. If $[\![\exists v_1, \dots v_n \phi]\!]_{\mathscr{A}} = T$, there is a variable assignment α such that $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = T$; but this variable assignment must assign $[\![\tau]\!]^{\alpha}_{\mathscr{A}}$ to each variable (there is only one element in the domain), so $[\![\phi[\tau/v_1, \dots, \tau/v_n]]\!]^{\alpha}_{\mathscr{A}} = T$, contradiction. (Using Theorem 104.) □

These ∃-sentences are *equi-decidable* with quantifier-free sentences.

Theorem 120 (Arbitrary ∃-sentences)

If ϕ is a quantifier free formula containing only $\tau_1, \dots, \tau_n$ and $v_1, \dots, v_m$, then

$$\vDash \exists v_1 \dots v_m \phi \quad \textit{iff} \quad \vDash \bigvee(\phi[\tau_i/v_j]),$$

where $\bigvee(\phi[\tau_i/v_j])$ is the disjunction of all *the ways of substituting the constants $\tau_1, \dots, \tau_n$ for the variables $v_1, \dots, v_m$.*

I leave the proof of this for an exercise; it is along the same lines as the proof of the proof for constant-free ∃-sentences.

Now we have shown that an arbitrary sentence involving only existential quantification is valid iff some quantifier-free sentence is valid, and since the latter is decidable, so is the former. Now we just need to show that any ∀∃-sentence is valid iff some ∃-sentence is valid.

∀∃-Sentences

Theorem 121 (∀∃-sentences and ∃-sentences)
If $\tau_1, \ldots, \tau_m$ don't occur in ϕ, then

$$\vDash \forall v_1 \ldots v_m \exists v_1 \ldots v_n \phi \quad \textit{iff} \quad \vDash \exists v_1 \ldots v_n \phi[\tau_1/v_1, \ldots, \tau_m/v_m].$$

Proof. L to R: By repeated applications of ∀-Elimination (Theorem 105), we have

$$\forall v_1 \ldots v_m \exists v_1 \ldots v_n \phi \vDash \exists v_1 \ldots v_n \phi[\tau_1/v_1, \ldots, \tau_m/v_m],$$

from which the result follows by Cut.

R to L: Because Γ is empty, and $\tau_1, \ldots, \tau_m$ don't occur in ϕ, we can use ∀-Introduction (Theorem 106) repeatedly to show the theorem. □

So any ∀∃-sentence is equi-valid with some quantifier-free sentence.

Elementary Quantifications and Monadic Sentences A sentence is an *elementary quantification* iff no quantifier occurs in the scope of any other. Some sentences won't be equivalent to any elementarily quantified sentence – e.g., $\forall x \exists y Rxy$ can't be put into that form.

An $\mathscr{L}_2$ sentence is *monadic* iff it contains at most unary (one-place) predicates.

Lemma 122 (Monadic and Elementary Sentences)
Every monadic sentence of $\mathscr{L}_2$ is equivalent to some elementary quantification.

Proof. Left as a problem. The general idea is: since we have no binary predicates, each bound variable in a sentence is associated with a unique quantifier, and we can 'drive in' the quantifiers so that it binds only its own variable. □

Corollary: Monadic $\mathscr{L}_2$ decidable

Theorem 123 (Monadic $\mathscr{L}_2$ sentences)
If ϕ is a monadic sentence of $\mathscr{L}_2$, it is equivalent to a $\forall\exists$-sentence.

Proof. Suppose ϕ is a monadic sentence. We can effectively produce a an elementary quantification ϕ', logically equivalent to ϕ. But switching the order of any quantifiers in that driven in sentence can't change the truth-value (exercise). And so we can reorder the quantifiers in ϕ' to get another logically equivalent sentence ϕ'' that is $\forall\exists$. Each step is effective: monadic $\mathscr{L}_2$ is decidable. □

Further Reading

Further discussion of the Harris Theorem is in McGee (2006).

The completeness proof was first established by Gödel in his 1929 doctoral thesis; the proof sketched here follows the later proof of Henkin (1949). The technique of associating (in)valid arguments with (un)defined computable functions mentioned in the proof of undecidability is involved. An elementary version is in Jeffrey (2006: ch. 7–8). A more sophisticated version is in Boolos *et al.* (2007: ch. 1–8). The connection between decidability and the halting problem was shown by Church and Turing, independently: Turing's result is in Turing (1937). This technique for proving decidability of $\forall\exists$-sentences is in Bostock (1997: §3.9). He provides an alternative decision procedure for monadic $\mathscr{L}_2$ in §3.8.

Exercises

1. Prove, using the hints in the chapter, the Cut Theorem: that if $\Gamma \vdash \phi$ and $\Delta, \phi \vdash \psi$, then $\Gamma, \Delta \vdash \psi$.
2. Explain why we could have used this rule instead of our ∀Intro:

$$\dfrac{\begin{array}{c}\vdots\\ \phi\end{array}}{\forall\upsilon\phi[\upsilon/\tau]}\ \forall\text{Intro}$$

Provided τ doesn't occur in any undischarged assumption in the proof of ϕ.

3. Let ψ be the result of substituting υ for *all* occurrences of τ in ϕ. Show that this rule is not sound (i.e., not truth-preserving)

$$\text{If } \Gamma, \phi \vdash \chi \text{ then } \Gamma, \exists\upsilon\psi \vdash \chi, \text{ provided } \tau \text{ does not occur in } \Gamma \text{ or in } \chi.$$

Can you think of a way of amending the rule to make it sound? *(Hint: Theorem 10.3.)*

4. Verify that all the rules of *ND* are sound with respect to the $\mathscr{L}_2$ semantics.

5. (a) Show that the rule ∃Intro is sound with respect to the $\mathscr{L}_2$ semantics.

   (b) Show that the rule ∀Intro is sound with respect to the $\mathscr{L}_2$ semantics.

6. Explain why one cannot define the relation 'x is an ancestor of y' in terms of the relation 'x is a parent of y'. What resources could be added to the language to allow this to be defined?

7. Sketch the proof that for any finite set of $\mathscr{L}_2$ sentences, we can effectively produce every proof that uses only sentences in that set.

8. Assuming that $\wedge, \neg, \vee$ are the only truth functional connectives in ϕ:

   (a) Show that, where v is not free in ψ, that $\psi \wedge \forall v\phi \equiv \forall v(\psi \wedge \phi)$.

   (b) Show that, where v is not free in ψ, that $\psi \wedge \exists v\phi \equiv \exists v(\psi \wedge \phi)$.

   (c) Show that, where v is not free in ψ, that $\psi \vee \forall v\phi \equiv \forall v(\psi \vee \phi)$.

   (d) Show that, where v is not free in ψ, that $\psi \vee \exists v\phi \equiv \exists v(\psi \vee \phi)$.

   (e) Show that $\forall v\phi \wedge \forall v\psi \equiv \forall v(\phi \wedge \psi)$.

   (f) Show that $\exists v\phi \vee \exists v\psi \equiv \exists v(\phi \vee \psi)$.

9. (a) Using the results in problem 8, plus the results on the interdefinability of quantifiers from week 5, prove the PNF theorem for the fragment of $\mathscr{L}_2$ without arrow or biconditional.

   (b) Prove the PNF in the following stronger form:

   > For every sentence ϕ in the arrow and biconditional-free fragment of $\mathscr{L}_2$, there is a logically equivalent sentence ϕ' in PNF *that has no more connectives than* ϕ.

   (c) Can we extend the stronger result to the full language $\mathscr{L}_2$?

10. Prove the Lemma on Quantifier-Free Sentences.

11. Prove the Theorem for Arbitrary ∃-Sentences.

12. Suppose that ϕ is a monadic $\mathscr{L}_2$ sentence containing only $\wedge, \vee, \neg$ as connectives in addition to quantifiers.

   (a) Show that we can construct a sentence ϕ' logically equivalent to ϕ in which no quantifier occurs in the scope of any other – i.e., an elementary quantification. (Consider exercises 8.(a–f), and recalling facts about DNF and CNF, thinking particularly about giving a CNF 'equivalent' of an open formula in DNF and *vice versa*.)

(b) Show that $\forall v\phi \wedge \forall v\psi \equiv \forall v\phi \wedge \forall v\psi[v/v]$ (and similarly for disjunction, and other combinations of universal and existential quantifiers).

(c) Show that for any elementary quantification ϕ' we can construct a logically equivalent sentence ϕ'' in which any quantifier in ϕ' is brought to the front and has scope over the others. (Tip: remember to use the result proved in 12.(b).)

(d) Show, using these results, that there is a $\forall\exists$ sentence equivalent to any such monadic ϕ.

13. Prove that there is no sentence ϕ which contains just one occurrence of some two-place connective, just one occurrence of any quantifier, which is in PNF and is equivalent to $P \leftrightarrow \forall x Q x$. Is there any connective other than $\leftrightarrow$ for which this also holds? Why?

14. The *Cantor Pairing Function* is this function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$:

$$\pi(\langle x, y\rangle) = \frac{(x + y)(x + y + 1)}{2} + y.$$

Show that this function has an inverse (i.e., a function from natural numbers to pairs of natural numbers). Show therefore that π is one-one and onto, and that the set of pairs of natural numbers is countable. (Hint: show that, given some z, we can reconstruct a unique x and y from it such that $\pi(\langle x, y\rangle) = z$.)

Chapter 13

Syntax, Semantics, and Derivations in $\mathscr{L}_=$

13.1 Identity

Identity: Syntax and Semantics The language $\mathscr{L}_2$ doesn't have a privileged predicate for identity. Yet such a predicate is very useful in discussing binary relations.

We *add* to $\mathscr{L}_2$ a binary predicate '='. We add new atomic formulae: where τ_1 and τ_2 are any terms, '$\tau_1 = \tau_2$' is an atomic formula. We keep the same structures as $\mathscr{L}_2$, but give this new predicate a constant semantic value, so that in every $\mathscr{L}_2$-structure $\mathscr{A}$,

(=) $$I_{\mathscr{A}}(=) = \{\langle x, x\rangle : x \in D_{\mathscr{A}}\}$$

Of course the property that we intend this predicate to express could exist in an $\mathscr{L}_2$-structure already; the change is that we now give it a logically privileged meaning. The satisfaction conditions are:

- $[\![\tau_1 = \tau_2]\!]^{\alpha}_{\mathscr{A}} = T$ iff $\langle [\![\tau_1]\!]^{\alpha}_{\mathscr{A}}, [\![\tau_2]\!]^{\alpha}_{\mathscr{A}}\rangle \in [\![=]\!]^{\alpha}_{\mathscr{A}}$, i.e., iff $[\![\tau_1]\!]^{\alpha}_{\mathscr{A}} = [\![\tau_2]\!]^{\alpha}_{\mathscr{A}}$; i.e., if $[\![\tau]\!]^{\alpha}_{\mathscr{A}}$ is the very same object as $[\![\kappa]\!]^{\alpha}_{\mathscr{A}}$.

Call the language which keeps all the old atomic formulae and formation rules

of $\mathscr{L}_2$, but adds these atomic formulae, $\mathscr{L}_=$.

Some Theorems About Identity

Theorem 124 (Theoremhood of Identity)
For any constant τ, $\vDash \tau = \tau$.

Proof. Suppose $\nvDash \tau = \tau$ for some τ. Then there is a structure $\mathscr{A}$ such that $[\![\tau = \tau]\!]_{\mathscr{A}} = F$, iff $[\![\tau]\!]^{\alpha}_{\mathscr{A}} \neq [\![\tau]\!]^{\alpha}_{\mathscr{A}}$; impossible. □

Theorem 125 (Substitution of co-designating terms II)
$\tau = \kappa \vDash \phi[\tau/\upsilon] \leftrightarrow \phi[\kappa/\upsilon]$.

Proof. Fairly immediate from the Substitution of Co-Designating Terms theorem (chapter 5, page 179).

The Relation of Identity

Theorem 126 (Identity an Equivalence Relation)

- $\vDash \forall x x = x$;
- $\vDash \forall x \forall y (x = y \rightarrow y = x)$;
- $\vDash \forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$.

Theorem 127 ('Leibniz' Law')
For any ϕ *in which at most* υ *occurs free,* $\vDash \forall \upsilon \forall \nu (\upsilon = \nu \rightarrow \phi \leftrightarrow \phi[\nu/\upsilon])$.

Proofs left for exercises. *Leibniz' Law* usually refers to this *definition* of identity in second order logic: $\forall \upsilon \forall \nu (\upsilon = \nu \leftrightarrow \forall \Phi (\Phi(\upsilon) \leftrightarrow \Phi[\nu/\upsilon](\nu)))$.

Proofs in $\mathscr{L}_=$ To obtain a natural deduction system for $\mathscr{L}_=$, $ND_=$, we *add* to the rules of ND_2 the rules in Table 13.1, keeping all other rules. In =Elim-l and =Elim-r, ϕ is a formula in which at most υ occurs free.

$$\dfrac{\dfrac{[\tau = \tau]}{\vdots}\ =\text{Intro}}{\dfrac{\phi[\tau/\upsilon] \quad \tau = \kappa}{\phi[\kappa/\upsilon]}\ =\text{Elim-r} \qquad \dfrac{\phi[\tau/\upsilon] \quad \kappa = \tau}{\phi[\kappa/\upsilon]}\ =\text{Elim-l}}$$

Table 13.1: New Natural Deduction Rules for $\mathscr{L}_=$

Dispensibility of =Elim-l/=Elim-r The two rules =Elim-l and =Elim-r obviously offer very similar resources in proofs. If we could show that $\tau = \kappa \vdash \kappa = \tau$ using just *one* of these rules, we could show the other to be dispensible.

Theorem 128 (Dispensibility of =Elim-r (or =Elim-l))
In the presence of the rule =Elim-l (respectively, =Elim-r), the functionality of =Elim-r (=Elim-l) can be derived.

Proof. $\dfrac{\phi[\tau/\upsilon] \qquad \dfrac{[\kappa = \kappa] \quad \tau = \kappa}{\kappa = \tau}\ =\text{Elim-l}.}{\phi[\kappa/\upsilon]}\ =\text{Elim-l}$

This is obviously equivalent to =Elim-r; a similar proof would show the dispensibility of =Elim-l. (Note the use of =Intro.) □

Soundness of $\mathscr{L}_=$

Theorem 129 ($\mathscr{L}_=$ is Sound)
If ϕ is a sentence of $\mathscr{L}_=$, then if $\vdash_{\mathscr{L}_=} \phi$ then $\vDash \phi$.

Proof. We only need show that proofs extended by the new rules are sound.

- We have a proof of ϕ on assumption $\tau = \tau$, and apply =Intro to discharge the assumption. Since by the Theoremhood of Identity, $[\![\tau = \tau]\!]_{\mathscr{A}} = T$ in every structure, and by the induction hypothesis $[\![\phi]\!]_{\mathscr{A}} = T$ if $[\![\tau = \tau]\!]_{\mathscr{A}} = T$, clearly $[\![\phi]\!]_{\mathscr{A}} = T$.

- We have a proof of $\tau = \kappa$ on assumptions Γ, and a proof of $\phi[\tau/\upsilon]$ on assumptions Δ, and we extend to a proof of $\phi[\kappa/\upsilon]$ using =Elim-r on assumptions $\Gamma \cup \Delta$. By the induction hypothesis, for any $\mathscr{A}$ where $\Gamma \cup \Delta$ is satisfied, $[\![\tau = \kappa]\!]_{\mathscr{A}} = T$, and similarly $[\![\phi[\tau/\upsilon]]\!]_{\mathscr{A}} = T$. By the Substitution of

co-designating constants theorem of page 201 and logic, in any such $\mathscr{A}$, $[\![\phi[\kappa/\upsilon]]\!]_{\mathscr{A}} = T$. □

Further metalogical results It turns out that the proof of completeness of $\mathscr{L}_2$ can be extended to provide a completeness proof for $\mathscr{L}_=$.

Theorem 130 (Completeness of $\mathscr{L}_=$)
If $\vDash \phi$ *then* $\vdash \phi$.

In the now familiar way, this can be adapted to a compactness proof.

Since $\mathscr{L}_2$ is not decidable, and $\mathscr{L}_=$ contains $\mathscr{L}_2$ as a part, $\mathscr{L}_=$ is not decidable either. Obviously the monadic fragment of $\mathscr{L}_=$ is decidable (as '=' is binary, the monadic fragment is the same as that of $\mathscr{L}_2$). A more interesting fragment is *the pure theory of identity* – the fragment where the only atomic formulae are of the form $\tau = \kappa$. This, it turns out, is decidable (Bostock, 1997: 329–31).

13.2 Numerical Quantification and the Theory of Definite Descriptions

Numerical Quantification: 'At Least' The ordinary interpretation of the sentence $\exists x Px$ is that there are one or more things which satisfy P. Obviously, $\exists x \exists y Px \wedge Py$ does not express that there are two or more things which satisfy P, because x and y may take the same values under a variable assignment. To express that there are at least two Ps, we need to guarantee the *distinctness* of the values of the variables. We can do this using identity:

$$\exists x \exists y((Px \wedge Py) \wedge \neg x = y).$$

Similarly,

$$\exists x \exists y \exists z((Px \wedge Py \wedge Pz) \wedge (\neg x = y \wedge \neg y = z \wedge \neg x = z))$$

expresses that there are *at least three* distinct Ps. Let $\exists_n \phi$ express that there are at least n things which satisfy ϕ.

Numerical Quantification: 'At Most', 'Exactly' Just as for 'at least', so we can formalise 'there are at most n Ps'. Under what circumstances are there at most n things? Just in case, if we chose $n + 1$ times, we would have to have chosen some of the things twice: some of things chosen would have to be identical. So we formalise 'there are at most 2 Ps' as

$$\forall x \forall y \forall z((Px \wedge Py \wedge Pz) \rightarrow (x = y \vee x = z \vee y = z)).$$

Formalise 'there are at most n ϕs' as $\forall_n x \phi$. Then we can say that *there are exactly* n ϕ as

$$\exists_n x \phi \wedge \forall_n x \phi.$$

There will be more or less concise ways of expressing this claim.

Definite Descriptions The $\mathscr{L}_=$ sentence Pa is often a good translation of simple English sentences of subject-predicate form. Yet some sentences, with this apparent form, cannot be translated in this way. Consider 'The fourth-oldest college is a college'. While 'the fourth-oldest college' is a referring expression, denoting a particular item in the domain, it seems a bad idea to formalise this by a constant a. For one thing, the English sentence is a tautology, but the proposed formalisation is not. So some English expressions which, like constants, denote an object in the domain, shouldn't be formalised by constants. It is notable that 'the fourth-oldest college' doesn't name a particular college (Exeter) – it *describes* it. A *definite description* designates a particular thing by giving a description that the thing – and *only* that thing – satisfies. We will look again at the relation between constants of $\mathscr{L}_2/\mathscr{L}_=$ and English referring expressions in Chapter 8.

So 'the strongest man in the world' designates that person who satisfies the property of being a man stronger than any other. 'Strongest' entails uniqueness, and this is generally true for definite descriptions: they fail to refer if more than one thing satisfies the description (so 'the college on Turl St' fails, because there are three such). *Indefinite* descriptions, such as 'a strong man', do not require uniqueness.

Definite descriptions also seem to require existence of the thing they describe:

the definite description 'the present king of France' fails because *nothing* satisfies that description.

Definite Descriptions and Numerical Quantification Putting these ideas together, we might say that the definite description 'the P' refers to the unique existing thing that is P. But we know how to express that there is one and only one P, using numerical quantification – so we can say that 'the P is Q', because we can say that there is one and only one P, and it is Q. Where ϕ, ψ are expressions in which at most v occurs free, let us introduce the notation $\imath v : \phi$ to denote 'the v such that ϕ', or simply, 'the ϕ'. One could permit this expression can take the place of a constant, so if $\psi\tau$ is well-formed, so is $\psi(\imath v : \phi)$; the latter says 'the ϕ is ψ'. Given our resources, though, we needn't add this to the language, or deal with the complications that ensue, for this expression can be defined:

Definition 113 (Definite Descriptions). $\psi(\imath v : \phi) =_{\text{df}} \exists v(\psi \wedge \forall v\phi[v/v] \leftrightarrow v = v)$.

This, essentially, is *Russell's analysis of the logical form of definite descriptions.*

Descriptions and Scope One reason to suspect that definite descriptions aren't just ordinary names – and a corresponding reason to reject the descriptivist account of the meaning of the constants of natural language – is that descriptions, unlike names, have *scope*. Consider

(*) The prime minister has always been Australian.

In $\mathscr{L}_=$, there is no non-truth-functional operator 'it always has been that ϕ'. We can mimic this (or maybe it is not mimicking) by *quantifying over times*. Let t be 'now', $\prec$ be 'is earlier than', P be 'is prime minister at', and A be 'is Australian'. (*) has two readings:

$$\forall x(x \prec t \rightarrow \exists y((Pyx \wedge \forall z(Pzx \leftrightarrow y = z))) \wedge Ay);$$
$$\exists y((Pyt \wedge \forall z(Pzt \leftrightarrow y = z)) \wedge \forall x(x \prec t \rightarrow Ay)).$$

The first says: always the prime minster, whoever they've been, has been Australian at the time of their prime ministership; the second says the present prime minister has always been Australian.

(Very similar results can be seen in *modal* contexts of possibility and necessity. We can again mimic the behaviour of the operator 'possibly' as an existential quantifier over 'possible worlds', and 'necessarily' (like 'always') as a universal quantifier.)

The problem is that in the case of names we seem *not* to see these scope problems. For this claim has only one reading:

(†) Kevin Rudd has always been Australian.

The existence of scopes for definite descriptions, is some evidence for the fact that they are really quantifier expressions.

Yet there are some disanalogies. Consider

(‡) The present king of France is bald.

According to Russell's theory, this is *false* – existence fails. But we may not share that intuition; this claim may strike us as neither true nor false, or having a false presupposition and thus unassessable.

Alternative Theories of Descriptions Strawson used our uncertain intuitions about (‡) to argue that descriptions are often used referentially,

> to mention or refer to some individual person or single object…, in the course of doing what we should normally describe as making a statement about that person [or] object.

When uttering a referring expression, one *presupposes* the existence of the thing. But when our intentions go awry, and we have presupposed something false, the utterance fails to have a truth value.

One truly weird way of dealing with (‡) is due to Meinong. This is to suppose that definite descriptions always pick out something, but sometimes the things picked out do not exist! Rather, according to Meinong, they *subsist*. Meinongianism apaprently handles one type of problematic definition description with ease:

(¶) The present king of France does not exist.

But one might legitimately query whether a real success has been achieved here. For Meinong, apparently empty definite descriptions, like the present king of

France, do function referentially: they pick out merely subsisting entities. But then Meinong offers us an explanation as to why we cannot use existential generalisation (the English analogue of the $\mathscr{L}_2$ rule $\exists$-intro) on such singular terms. For we manifestly cannot use it, else the true claim (¶) would entail this false claim:

(!) There is something that does not exist.

13.3 Compactness and Cardinality

'There are *n* things' and infinity We saw before that there is a set of $\mathscr{L}_=$ sentences satisfiable only in infinite domains. If R is a transitive asymmetric relation (both of which properties are expressible by an $\mathscr{L}_=$ sentence), then $\forall x \exists y Rxy$ will be satisfiable in no finite domain. Can we give a set of sentences satisfiable in *all and only* infinite domains?

Using the resources for numerical quantification, define $\mathbf{n}$ as $\exists_n x(x = x)$. Then $\mathbf{3}$ says there are at least three things.

Consider the set $\mathbf{N} = \{\mathbf{n} : n \in \mathbb{N}\}$. $\mathbf{N}$ is not satisfiable in any finite domain: suppose it were satisfiable in domain of size m, then where $n = m + 1$, $\mathbf{n}$ would be false and because $\mathbf{n} \in \mathbf{N}$, $\mathbf{N}$ is not satisfied. So $\mathbf{N}$ is satisfiable only in infinite domains. Moreover, in every infinite domain, $\mathbf{N}$ is satisfied. (exercise).

$\mathbf{N}$ is thus an 'axiomatisation of infinity'. It is a consequence of compactness that there is no single $\mathscr{L}_=$ sentence which is an axiom of infinity (exercise).

Finitude Undefinable Since there is at least one sentence of $\mathscr{L}_=$ true only in infinite domains, perhaps there is a sentence of $\mathscr{L}_=$ true only in *finite* domains? Perhaps surprisingly, there is not.

Theorem 131 (Finitude Undefinable)
There is no set of $\mathscr{L}_=$-sentences satisfiable in all and only finite domains.

Proof. Suppose FIN is a set of $\mathscr{L}_=$ sentences true in all and only finite domains. $FIN \cup \mathbf{N}$ must be unsatisfiable; by compactness, where Γ is a finite subset of FIN and Δ is a finite subset of $\mathbf{N}$, $\Gamma \cup \Delta$ is unsatisfiable. As Δ is a finite subset of $\mathbf{N}$, there is some greatest $\mathbf{n} \in \Delta$; in any model $\mathscr{A}$ with a domain of size greater than n, Δ is satisfiable in $\mathscr{A}$. Since $\Gamma \cup \Delta$ is unsatisfiable, Γ is unsatisfiable in any such

model $\mathscr{A}$, so there is a model with a finite domain (any such $\mathscr{A}$) in which FIN is unsatisfiable, contrary to assumption. So there is no such FIN.

Löwenheim-Skolem Theorem

Theorem 132 (The Löwenheim-Skolem Theorem)
If Γ is a set of sentences of $\mathscr{L}_=$ which has an infinite model $\mathscr{A}$ (of size ω_α), then

Upward *Γ has a model for every infinite size $\omega_\beta > \omega_\alpha$.*

Downward *Γ has a countable model.*

Proof. Upward: Extend the language for Γ to $\mathscr{L}^+$ by adding new constants $\boldsymbol{C}$ such that size of $\boldsymbol{C}$ is ω_β. Consider the set of $\mathscr{L}^+$ sentences $\Gamma^* = \Gamma \cup c_i \neq c_j : i \neq j, c_i, c_j \in \boldsymbol{C}$. Since Γ is satisfiable, every finite subset of Γ^* is satisfiable – by compactness (for this uncountable language!), Γ^* has a model, and the model must have cardinality ω_β.

Downward: too complex to be covered here, unfortunately – see, e.g., Boolos *et al.* (2007: chs. 12–13). □

Overspill

Theorem 133 (Overspill)
If a set of sentences has an arbitrarily large finite model, it has a countable model.

Proof. Let Γ have arbitrarily large finite models. Let $\Gamma' = \Gamma \cup \mathbf{N}$. Any finite subset of Γ' is a subset of $\Gamma \cup \Delta$, where Δ is a finite subset of $\mathbf{N}$, with a greatest $\mathbf{n} \in \Delta$. Since Γ has arbitrarily large models, $\Gamma \cup \Delta$ has a model. But then every finite subset of Γ' has a model; by Compactness, Γ' has a model. And by the downward Löwenheim-Skolem theorem, it has a countable model (Boolos *et al.*, 2007: 147).) □

Skolem's Paradox The Löwenheim-Skolem theorem says that no first-order theory can restrict itself to models of one infinite cardinality. But set theory is a first-order theory: the Löwenheim-Skolem theorm says it has a (presuambly unintended) countable model, but set theory itself entails that there are uncountable sets! This is *Skolem's paradox*.

The resolution of the 'paradox' is to note that set theory

> only "says" that [some set] S is nondenumerable in a *relative* sense: the sense that the members of S cannot be put in one-to-one correspondence with a subset of $\mathbb{N}$ by any R *in the model*. A set S can be "nondenumerable" in this *relative* sense and yet be denumerable "in reality". This happens when there *are* one-to-one correspondences between S and $\mathbb{N}$ but all of them lie outside the given model. What is a "countable" set from the point of view of one model may be an uncountable set from the point of view of another model. (Putnam, 1980: 465)

13.4 Further Directions of Research and Study in Logic

We could now proceed to extend, or alter our logic, in quite a few ways:

Enriching We could add still further to $\mathscr{L}_=$ or $\mathscr{L}_1$, introducing *modal operators* like 'necessarily' and 'possibly', *temporal operators* like 'it was the case that', or even moral operators like 'it ought to be that', or stronger conditionals like the *counterfactual* 'If it had been the case that ϕ, it would have been the case that ψ'. Then we can prove results about these enriched systems.

Altering Some complain that $\neg\neg\phi \nvDash \phi$; these *intuitionists* want to revise our logic. *Free logic* permits constants to lack a designation in a model, and *universally free logic* abandons the requirement that the domain be non-empty; these of course will also revise our logic as given.

Advancing We could augment our languages in another way too, perhaps introducing *function symbols* – operators on terms that yield terms – like + or ×. We can then formalise languages of interest – most famously *arithmetic*, in the *Peano axioms*, and prove various things about them. In this way we can build up to one of the most celebrated results of twentieth century logic:

Theorem 134 (Gödel Incompleteness Theorems)
If T is a consistent, finitely axiomatisable theory able to express elementary arithmetic, then

- *T cannot prove every arithmetical truth;*

- *T cannot prove its own consistency (i.e., cannot prove an arithmetical sentence that is true iff T is consistent). (This second result obviously entails the first given that T is consistent.)*

The proof is sadly beyond us.

In the next chapter, we will look at some of the philosophical issues arising for the logic we have introduced. In particular, we will look at how well the operator $\rightarrow$ matches the English conditional, and how well the semantics of designators and relations for $\mathscr{L}_2$ matches similar constructions in English. We'll also reexamine the sequent $\phi, \neg\phi \vdash \psi$.

Further Reading

Russell's famous paper on definite descriptions is Russell (1956); the nearly as famous response by Strawson is Strawson (1950). A good overall source is Ludlow (2013).

One much-discussed philosophical analysis of the consequences of the Löwenheim-Skolem theorem is Putnam (1980). This article is very controversial! One response is Lewis (1984).

On modal, temporal, etc., logics, a good treatment is by Beall and van Fraassen (2003); a very elegant recent examination of these topics an the topicof non-classial logic generally is Burgess 2009. Free logics are treated by Bostock (1997: §§8.4–8.7), and Lambert (2001).

A good text on the further development of metamathematics is Boolos *et al.* 2007.

Exercises

1. Prove the theorem called 'Substitution of co-designating constants' from page 201.
2. Prove the following
   (a) $\vDash \forall x\, x = x$;
   (b) $\vDash \forall x \forall y (x = y \rightarrow y = x)$;
   (c) $\vDash \forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$.
3. (a) Show that in the presence of =Elim, $\exists$Elim and $\exists$Intro, the rule =Intro can be replaced by this rule (where τ is a constant):

$$\dfrac{\begin{array}{c}[\exists \upsilon\, \upsilon = \tau] \\ \vdots\end{array}}{} \text{=Intro}\dagger$$

(b) What would be the effect of adopting, instead of our rule =Intro, this rule (in the presence of the other rules of ND_2):

$$\frac{\phi}{\tau = \tau}\ =\text{Intro}^*$$

(c) Can we replace both =Intro and =Elim by this pair of rules, in the presence of the other rules of ND_2 (ϕ is a formula in which at most υ occurs free):

$$\frac{\phi[\tau/\upsilon]}{\exists\upsilon(\upsilon = \tau \wedge \phi)}\ =\text{Intro}‡ \qquad \frac{\exists\upsilon(\upsilon = \tau \wedge \phi)}{\phi[\tau/\upsilon]}\ =\text{Elim}‡$$

4. Suppose ϕ is a formula in which at most υ occurs free, and in which ν does not occur at all.

   (a) Show that $\vDash \forall\upsilon\forall\nu(\upsilon = \nu \rightarrow \phi \leftrightarrow \phi[\nu/\upsilon])$.

   (b) Suppose for *every* formula ϕ with just υ free, not itself involving identity, and any constants τ, κ, we could show that $[\![\phi[\tau/\upsilon]]\!]_{\mathscr{A}} = [\![\phi[\kappa/\upsilon]]\!]_{\mathscr{A}}$. Could we then show that $[\![\tau = \kappa]\!]_{\mathscr{A}} = T$? (This is intended to mimic as best we can the second-order quantification over properties in the definition of identity: the question is, does it work in the right way?)

5. (a) Show that $\exists_{n+1} x Px \equiv \exists x(Px \wedge \exists_n y(Py \wedge \neg x = y))$.

   (b) Define $\forall_{\bar{n}} x\phi$ as $\neg\exists_n x\neg\phi$. Give an English translation of $\forall_{\bar{n}} xPx$.

6. In the analysis of definite descriptions, $\exists\upsilon(\psi \wedge (\forall\nu\phi[\nu/\upsilon] \leftrightarrow \upsilon = \nu))$, could we replace the '$\leftrightarrow$' by '$\rightarrow$'? If not, why? What about in this formulation: $\exists\upsilon(\psi \wedge (\forall\nu\upsilon = \nu \leftrightarrow \phi[\nu/\upsilon]))$?

7. Give appropriate analyses in $\mathscr{L}_=$ of these claims, being sure to point out any cases of ambiguity, and commenting on other issues:

   (a) The number of planets is necessarily eight.

   (b) The prime minister must be an honest person.

   (c) There used to be a Labor prime minister.

8. What is the best way to formalise 'x exists' in $\mathscr{L}_=$? What are the difficulties with a Russellian analysis of (¶) 'The present king of France does not exist'?

9. Can the Russellian analysis of descriptions deal with the following sentences?

   (a) 'The beer is all gone.'

   (b) 'He is tall, handsome, and the love of my life.'

(c) 'She is the proud owner of a Rolls Royce.'

10. Can we use identity and quantification to express 'there are exactly as many Ps as Qs'?

11. (a) Show that the infinite set of sentences $\mathbf{N}$ is satisfied in every infinite domain.

(b) Show that there is no single $\mathscr{L}_{=}$-sentence which is satisfied in $\mathscr{A}$ iff $\mathbf{N}$ is satisfied in $\mathscr{A}$.

12. A *permutation* on S is a one-one function π from S to S. If $\mathscr{A}$ is a model, where π is a function from the domain of $\mathscr{A}$ to itself, let the permutation $\pi(\mathscr{A})$ be this model

- $D_{\mathscr{A}} = D_{\pi(\mathscr{A})}$.
- For any term τ, $I_{\pi(\mathscr{A})}(\tau) = \pi(I_{\mathscr{A}}(\tau)$.
- For any n-ary predicate Φ,

$$I_{\pi(\mathscr{A})}(\Phi) = \{\langle \pi(x_1), \ldots, \pi(x_n)\rangle : \langle x_1, \ldots, x_n\rangle \in I_{\mathscr{A}}(\Phi)\}.$$

(a) Show the *Permutation Theorem*: for any sentence ϕ, and permutation π, $[\![\phi]\!]_{\mathscr{A}} = [\![\phi]\!]_{\pi(\mathscr{A})}$.

(b) Consider this passage:

> the Permutation Theorem shows that, even if we can somehow discount the problems raised by the [Skolem's paradox and the Löwenheim-Skolem Theorem], and settle on models with one fixed domain as those which can truly claim to be abstractions of the world – even then, there are as many different reference relations holding between the subsentential expressions of T's language and the world as there are nontrivial permutations of the fixed domain, all of which underwrite the truth of exactly the same sentences, and, hence, equally subserve the truth of the theses of T. There being no basis for preference between these reference relations, realism's doctrine that there is a unique, privileged such relation is apparently discredited. (Taylor, 2006: 53–4)

Assess and evaluate this argument. Do you accept the conclusion? If not, why not?

Answers to selected exercises on page 251.

Part V

Beyond Classical Logic

Chapter 14

Modal and Temporal Logic

Chapter 15

Logic and Natural Language Conditionals

15.1 Indicative Conditionals

'If ϕ, ψ' and $\rightarrow$ Focus – for now – on the English conditional construction 'If ϕ, ψ' where ϕ and ψ are both sentences in a simple tense (i.e., 'they walk', 'I walked', 'You will walk') For example: 'if you walk three blocks, you'll see it on your left'; or 'If Australia was already inhabited, Cook didn't discover it'. Such conditionals have been called *indicative* conditionals in the literature, because their simple tensed constituents are sometimes said to be in the 'indicative mood'.

When translating English into our formal languages $\mathscr{L}_1$ and $\mathscr{L}_2$, we have all been taught to render indicative conditionals using the *material conditional* $\rightarrow$. You may be worried, as others have worried before you, that this is not an accurate rendering. For, you may think, it is implausible that 'If ϕ, ψ' is true whenever ψ is true; and no more plausible to think that 'if ϕ, ψ' follows just from the falsity of ϕ. Consider these two arguments:

(7) a. The Tigers will not lose every game this season;
    b. Therefore: If the Tigers lose every game this season, they will make it to the finals.

(8) a. The Tigers will make it to the finals;

b. Therefore: If the Tigers lose every game this season, they will make it to the finals.

Supposing that 'if ϕ, ψ' expresses the same proposition as $\phi \to \psi$, both of these arguments should be valid. The first is an instance of the valid form $\neg\phi \vDash \phi \to \psi$, and the second is an instance of the valid form $\psi \vDash \phi \to \psi$. But neither of these arguments strike us as valid: even if we were in a good position to assent to the premises, we should be reluctant to assent to the conclusion. That reluctance is some evidence of invalidity (though perhaps not itself conclusive).

Can we do better in our logic? The first question we should address is: is there a better option in $\mathscr{L}_1$? And it seems that there is not. Suppose that 'If ϕ, ψ' is expressed by some truth function $f(|\phi|, |\psi|)$. One thing is definitely true; if $|\phi| = T$ and $|\psi| = F$, we want $f(|\phi|, |\psi|) = F$. This leaves us with 8 possible two-place truth-functions. Another thing we want is *asymmetry* – 'if ϕ, ψ' should not always have the same truth value as 'if ψ, ϕ'. That is, there exists at least one structure $\mathscr{A}$ where $f(|\phi|_{\mathscr{A}}, |\psi|_{\mathscr{A}}) \neq f(|\psi|_{\mathscr{A}}, |\phi|_{\mathscr{A}})$. This rules out four further functions (it rules out all those where $f(F, T) = F$), leaving us with four. Since we've just suggested in our discussion of the arguments above that it is undesirable for the truth of a conditional to follow just from the truth of the consequent or from the falsity of the antecedent, it would surely be even worse if the conditional were *equivalent* to the consequent or the negation of the antecedent. This rules out two of our remaining truth functions, leaving us with just two: $\to$ and a function g (expressed by $\neg(\psi \to \phi)$). But $g(T, T) = F$, hardly what we want from a conditional. So the only truth-function which satisfies minimal conditions necessary to count as a conditional is $\to$.

Meaning as characterised by derivation rules Of course, this is not a strong argument – that $\to$ is better than other candidates is no argument that $\to$ is itself adequate to give the truth conditions of the indicative conditional! But we can offer better considerations. Given that our derivation system for $\mathscr{L}_1$ is sound and complete, there is a sense in which those rules – collectively – govern or fix the meaning of the connectives. No other rules are necessary to ensure the right

things are provable, the things which correspond exactly to the meaning of the connectives.

Things may not be much different in natural languages. Of course we don't have soundness and completeness proofs, because we have neither a fully spelled out semantics nor a formal derivation system for natural language. But we do have characteristic patterns of inference for the connectives of English. Some of these patterns of inference look remarkably like those for the formal connectives of $\mathscr{L}_1$, notably the rules governing 'and'. (This is some motivation behind the nomenclature of 'natural deduction' – it is natural in the sense that it formalises intuitively fundamental characteristic patterns of inference for the corresponding operators.)

Whatever else we might want to say about the English indicative conditional 'if ϕ, ψ', it seems *prima facie* it should obey these rules:

Modus Ponens If you have established ϕ, and you have established 'if ϕ, ψ', then you can on that basis establish ψ.

Conditional Proof If you can establish ψ, conditional on the assumption that ϕ and perhaps some supplementary assumptions Γ, then you can establish 'if ϕ, ψ' solely on the basis of the assumptions in Γ.

It is obvious that something which obeys these rules looks very much like a conditional. A conditional helps us neatly summarise reasoning from assumptions, store it, and then use it when those assumptions come true – Conditional Proof captures the first part, and *modus ponens* the last.

But if these two rules hold of the English conditional 'if', then we can offer an argument that the English conditional is $\rightarrow$. More precisely, the argument is that 'If ϕ, ψ' is true iff $\phi \rightarrow \psi$ is:

> *Only If*: Suppose that if ϕ, ψ. Assume ϕ. By *modus ponens*, ψ. By $\rightarrow$Intro, $\phi \rightarrow \psi$, discharging the assumption.
>
> *If*: Suppose that $\phi \rightarrow \psi$. Assume ϕ. We can now derive ψ, by $\rightarrow$Elim. But we have now established ψ on the basis of the assumption ϕ, together with the supplementary assumption $\phi \rightarrow \psi$. By Conditional Proof, we can establish that if ϕ, ψ on the basis of the supplementary assumption $\phi \rightarrow \psi$ alone.

There is another argument for the *If* direction, the 'Or-to-If' argument (Stalnaker, 1975). Recall that the material conditional $\phi \rightarrow \psi$ is logically equivalent to the disjunction $\neg\phi \vee \psi$. But the truth function expressed by $\vee$, **a**, is also expressed by English 'or'. So from $\phi \rightarrow \psi$ we can conclude that either $\neg\phi$ or ψ. So if it turns out that ϕ, then ψ (otherwise we have neither $\neg\phi$ nor ψ). (You may detect a disguised instance of conditional proof lurking in the background of this argument.)

Replying to the apparent counterexamples – Grice If the above argument works, we're wrong to think that the cases where $\rightarrow$ doesn't seem to behave like 'If…' are genuine counterexamples. The problematic argument forms above are in fact valid, so our inclination to reject the inferences must be explained in some other way.

The explanation offered is that, even though the conditional conclusions of these arguments are true, they *suggest* something false, and that false suggestion is what we are responding to when we find the arguments problematic. The notion of 'suggestion' in play here has been precisely treated using Grice's notion of *conversational implicature*. Grice noted that many utterances seem to communicate more than what is strictly said, and gave a series of principles that he argued govern what is communicated by an utterance, based on what strictly it means (Grice, 1989). One classic example is this: 'some philosophers have beards'. An utterance of this sentence conversationally implicates that not all philosophers have beards, and that is what something most people will assume the speaker to be committed to in their utterance. Yet this is not a consequence of the utterance: 'some philosophers have beards; in fact, all of them do!' can perfectly well be true, and is no contradiction.

The Maxim of Quantity The principle Grice invokes to explain this implicature is his 'Maxim of Quantity' – simply put, this is the principle that cooperative speakers be as informative as they can be. A hearer, without good reason to think otherwise, will assume that a speaker is being cooperative, and hence that when the speaker utters a claim, it is also the most informative claim the speaker is in a position to responsibly utter. One measure of informativeness is this: If ϕ entails ψ, then ϕ is at least as informative than ψ (for it carries the information

that ψ, and perhaps additional information too if ψ does not entail ϕ). Since 'All philosophers have beards' entails 'some philosophers have beards', someone who utters the latter utters something less informative than they could utter, if in fact they thought that all philosophers have beards. Since they are cooperative and did not utter the stronger claim, therefore, the hearer infers that the speaker did not believe the stronger claim. So the hearer now knows that the speaker must believe that some but not all philosophers have beards, and the speaker therefore communicates that not all philosophers have beards to the hearer. Here is another example:

> Now, in ordinary life we expect each other to be at least moderately generous with our information. So if you ask me where John is, and I say 'either in Oxford or in London', you will tend to take for granted that I am giving you as much information as I relevantly can. If you later discover that I knew at the time that John was in Oxford … you will think that I was mildly dishonest. For I allowed you to believe that I did not know whether he was in Oxford or not; although I was perhaps, in a casuistical kind of way, careful not to say, in so many words…. But this much is obvious: if I say that John is either in Oxford or in London, and say nothing more, then, if John is in fact in Oxford, I have not said anything false, since what I do say is compatible with all the relevant facts.(Thomson, 1990: 67)

Applying Quantity to conditionals This last example is pertinent to our present concerns. The material conditional is equivalent to a disjunction $\neg\phi \vee \psi$. Any disjunct of a disjunction entails the disjunction, so contains more information. So an utterance of a disjunction implicates that neither disjunct is believed by the speaker. As Thomson put it,

> In saying 'if p then q' a speaker will say something which is in general anyway true or false. But by the act of making the statement he will do other things, too. He will encourage us to think that he has some or other reason for thinking that if p then q and that his reasons are not such as to allow him to assert not-p nor such as to allow him to assert q. (Thomson, 1990: 67–8)

So if it is apparent that the speaker *does* believe either disjunct – either through being committed to the consequent, or rejecting the antecedent, of the relevant material conditional – we should be struck by the fact that the speaker has communicated an apparent contradiction.

In the case of the Tigers, the speaker who utters both the premises and conclusion of this argument communicates that they believe the consequent, and also utters the conditional which conversationally implicates that the speaker does not believe the consequent. So the hearer has a contradiction communicated to them – little wonder, then, that these sentences sound so bad! But – and this is the crucial part – nothing in this explanation of the badness of the conditional is *semantic*. Nothing in this explanation undermines the idea that the truth conditions of the English conditional construction are the same as those for the material conditional. This is all to do with what we standardly take ourselves to be able to infer from an utterance of a sentence with those semantics, and what goes wrong here is that inference, not the semantics. If so, the truth conditions of 'if' might be those of $\rightarrow$. What makes it seem as though they are distinct is that, because sometimes it is *conversationally inappropriate* to say some things even though they are true, we get a strong negative reaction to certain true conditionals – and we get no corresponding reaction to to sentences involving $\rightarrow$, at least in part because those sentences are not used in ordinary conversations and so no rules have sprung up governing how they are to be used.

Trickier Cases All is not smooth sailing for the material conditional account of 'If ϕ, ψ'. I mention two trickier cases:

1. There seem to be cases where conditional proof goes wrong – most noticeably, in the case of assumptions made that explicitly contradict other things we believe. This is especially apparent in so called *counterfactual* conditionals – those (at least at first glance) that involve conditional reasoning about circumstances that do not obtain. So consider the (apparently true) counterfactual 'If Univ had never existed, Exeter would have been the third oldest college'. The conditional proof rationale for this is: if we can derive the consequent from the assumption that Univ never existed and other things, then we can derive the conditional from those other things alone. Yet since

Univ's existence is known to us, adding that assumption to our prior beliefs gives an inconsistent set of assumptions, from which any consequent whatever follows. But 'If Univ had never existed, Catz would have been the third oldest college' is clearly false, though it should be derivable on exactly the same grounds. What we need to do, apparently, is modify which of our prior beliefs can be retained when reasoning from this contrary-to-fact assumption.

Many philosophers are agreed, however, that such cases should be treated by analysing the 'if' involved in counterfactuals as distinct from the 'if' of the so-called *indicative* conditionals we've been dealing with so far – if that suggestion is adopted, we may retain a material conditional analysis of indicative 'if' while giving an alternative treatment of counterfactual 'if'. The classic early account of counterfactual 'if' is Lewis (1973).

2. There seem to be cases where the Gricean explanation should make a conditional sound bad, but in which it does not. Consider: 'You won't fail EDL. Even if you do, you will pass on the re-sit.' The first sentence of this little speech is a denial of the antecedent of the following conditional, so the conditional should sound bad. But it sounds fine. The word 'even' may be held responsible; but it is now up to the Gricean to explain how it interferes with the ordinary Gricean mechanisms.

The Gibbard Argument So far, we've restricted our attention to conditionals without conditional constituents. Conditionals with conditional antecedents are fairly rare in natural language, but there are many examples of conditionals with conditional consequents:

(9) a. If they were outside, then if it rained, they got wet (Edgington, 2014: §2.5);
b. If that's a fish, then if it has lungs, it's a lungfish (McGee, 1985);
c. If you drink one more can of beer then if I drink one more can of beer then we'll be completely out of beer (Kratzer, 2012: 88).

It appears, however, that these sort of right-nested conditionals are equivalent to un-nested conditionals with conjunctive antecedents. Compare these examples to those just above:

(10) a. If they were outside and it rained, they got wet;

b. If that's a fish and it has lungs, it's a lungfish;

c. If you drink one more can of beer and I drink one more can of beer then we'll be completely out of beer.

What seems plausible for these examples has been elevated to a general principle that a conditional with a conditional consequent is always equivalent to a conditional with a conjunctive antecedent.

Import-Export 'If ϕ then if ψ, χ' and 'If ϕ and ψ, then χ' are logically equivalent.

Suppose we accept Import-Export as a general principle about conditionals. Then we have the makings of another argument for the material conditional account of 'If'.

The argument begins with a technical result about any conditional operator.

Theorem 135 (Gibbard)
Any conditional operator $\Rightarrow$ which satisfies these three conditions is equivalent to the material conditional (Gibbard, 1981):

Import-Export $\phi \Rightarrow (\psi \Rightarrow \chi)$ *and* $(\phi \wedge \psi) \Rightarrow \chi$ *are logically equivalent.*

Lower $\phi \Rightarrow \psi$ *entails* $\phi \rightarrow \psi$ *(i.e., the conditional entails the material conditional).*

Upper $\phi \vDash \psi$ *entails* $\phi \Rightarrow \psi$ *(i.e., entailment entails the conditional)*

Proof. Left for exercise. □

The relevance of the theorem is that the English conditional 'if' does seem to satisfy Upper, Lower, and Import-Export. The only mildly controversial one is apparently the latter, but we've already seen how plausible it is in particular cases,

The Trickiest Case Here's another problem case:

> Opinion polls taken just before the 1980 election showed the Republican Ronald Reagan decisively ahead of the Democrat Jimmy Carter, with the other Republican in the race, John Anderson, a distant third. (McGee, 1985)

There are only two Republicans in the race, so this is trivially true:

(11) If a Republican wins and it's not Reagan, then it's Anderson who wins.

(11) is a conditional with a conjunctive antecedent; if Import-Export is valid for the English 'if', then we can derive

(12) If a Republican wins the election, then if it's not Reagan who wins it will be Anderson.

And since the polls overwhelmingly favour Reagan, it is almost certain he will win, and so almost certain that the antecedent of this conditional (12) - i.e., 'A Republican will win the election' – is true. It was certainly widely believed by people at the time that a Republican would win, and it turned out to be true in the end.

Assuming that if someone believes a conditional, and believes the antecedent of that conditional, they can validly reason (using *modus ponens*) to the consequent, this follows:

(13) If it's not Reagan who wins, it will be Anderson.

But this is false: Anderson was a distant third in the polls, and the most likely alternative to Reagan was Carter.

What has gone wrong? We've derived, from claims we believe to be true ((11) and that a Republican will win), using Import-Export and *modus ponens*, a claim we believe to be false: 13. The defender of the material conditional account of 'if' cannot object to either of these rules, since both Import-Export and *modus ponens* are provably correct derivation rules for the material conditional. So they have to argue that the apparently false conclusion is in fact true; or else deny either the trivial truth (11), or deny that it is acceptable to reason from the highly probable claim – that in fact turned out to be true, just as the polls indicated – that a Republican would win the election. None of these options looks particularly appealing, and most philosophers and linguists have concluded that the material conditional account of 'if' is thereby refuted. However, there has been no consensus about what to offer in its place.

Further Reading

Beall and van Fraassen (2003) and Burgess (2009) provide useful material on relevant and conditional logics. On the topic of the English indicative conditional, a good guide is Edgington (2014). In addition to the work of Grice and Thomson cited above, Davis (2014) is a useful source on conversational pragmatics. Stalnaker (1975) gives an account of the English conditional which invalidates Import-Export; McGee (1985) argues that in fact the English 'if' doesn't satisfy *modus ponens*. Edgington and Kratzer (2012) discuss more exotic accounts.

Exercises

1. Give the proof of Gibbard's theorem (Theorem 135). (*Hint:* begin with the fact that all instances of this schema for disjunctive syllogism are valid: $((\neg\phi \vee \psi) \wedge \phi) \vDash \psi$.)

2. (a) John argues that, since ϕ and ψ together entail (in English) ψ, if follows that ψ entails 'if ϕ, ψ'. Evaluate John's argument.

   (b) Mary claims that 'if ϕ, ψ' entails 'if $\neg\psi$, $\neg\phi$'. Evaluate Mary's claim.

   (c) Show how it would be possible to use John's conclusion and Mary's claim to argue that '$\phi \rightarrow \psi$' entails 'if ϕ, ψ'.

   (d) Give an argument that 'if ϕ,ψ' is a truth-functional connective in English. Do you see any difficulties with your argument?

Chapter 16

Logic and Natural Language: Entailment, Designators, and Relations

16.1 Entailment

Another problematic feature of $\mathscr{L}_1$ (somewhat related to the foregoing by the deduction theorem) are the so-called *paradoxes of entailment*. These are the following valid sequents:

- $\phi, \neg\phi \vDash \psi$;
- $\phi \vDash \psi \vee \neg\psi$.

But, some have objected, these are terrible inferences – we should never conclude, on the basis of contradictory premises, anything; and while it may be safe to conclude a tautology from any premises, that is hardly a good argument to persuade someone that the conclusion is a tautology.

Relevance Critics of these sequents have maintained that the failure here is one of *relevance*. The arguments may be valid in the narrow sense, because there is

no structure which satisfies the premises without also satisfying the conclusion (in the former case, because no structure satisfies the premises; in the latter, because every structure satisfies the conclusion). But validity in this narrow sense is too narrow – what we ordinarily understand to be a successful valid argument excludes these trivial cases of validity. The conclusion in a *genuinely* valid argument – these critics say – should follow from, and be relevant to, the premises.

The Lewis Argument Yet there is a very simple argument, using ordinary English reasoning which is apparently impeccable, to support these 'problematic' sequents. It is known as the *Lewis argument*, after C. I. Lewis, one of its modern rediscoverers.

(14)	ϕ	Assumption
(15)	ϕ or ψ	Disjunction introduction, 1
(16)	$\neg\phi$	Assumption
(17)	ψ	Disjunctive syllogism, 2, 3

'Disjunctive syllogism' is the rule that from $\phi \vee \psi$ and $\neg\phi$ one can infer ψ. (A problem asks you to show that this is valid derived rule in the system $\mathscr{L}_1$; it is obviously a good rule for the English 'or'.)

This argument seems to show that, even ordinarily, this form of apparently irrelevant argument is valid. And who could object to disjunction introduction or disjunctive syllogism, which seem like about the most basic things one can safely say about disjunction.

The relevantist response – reject Disjunctive Syllogism The relevantist has a response to this argument – reject disjunctive syllogism. For, they say, it goes wrong in precisely this circumstance. The inference works normally because, given a true disjunction with one false disjunct, we can infer to the truth of the other disjunct *without* checking to see whether it is true. (Disjunctive syllogism was called 'the dog' by its Stoic inventors, because 'even a dog uses this form of inference when it comes to a fork in the road, sniffs down one branch, and not finding the scent there immediately takes off down the other branch, without stopping to sniff' (Burgess, 2009: 99–100).) But in this case, ϕ is also an assumption – so the

fact that $\phi \vee \psi$ has a disjunct whose denial is an assumption doesn't entail ψ. If there are cases where we can have inconsistent assumptions, like this, we should expect disjunctive syllogism to go wrong.

Dialethism Some truly radical support for this line of argument can be found from the *dialethist* position that some sentences are *both* true and false. If ϕ is one such sentence, then both assumptions can be true (ϕ because it is true, $\neg\phi$ because ϕ is false). But ψ, since it is arbitrary, can be chosen to be a plain falsehood, giving a situation which is a counterexample to disjunctive syllogism. Dialethists motivate their non-standard semantics by appeal to particular problem cases that don't seem to admit of any other solution. The best known is

L L is false.

The sentence L displayed above says of itself that it is false. If it is true, then L is false (as that is what it says). But if it is false, then what it says is false, so it must be true. So L is true iff it is false. The dialethist takes this proof at face value – L is both true and false. But, despite the great difficulties more standard approaches to the Liar paradox L have, few have found dialethism compelling (though it has surprisingly able defenders).

Inconsistent belief Dialethism is too incredible to convincingly motivate the rejection of disjunctive syllogism. A weaker motivation is found in reasoning about what we believe. If you are like me, you probably have some inconsistent beliefs that you are not aware of. It is absurd to say that I therefore believe everything, just because I have beliefs that entail everything. So it seems, the set of my beliefs is not plausibly closed under classical entailment (that is, the set of my beliefs does not contain everything which is entailed by some things in my set of beliefs). Disjunctive syllogism seems to fail for my beliefs; just because $\phi \vee \psi$ and $\neg\phi$ are among my beliefs does not mean that ψ is one of my beliefs; for ϕ may be one of my beliefs as well. The logic of belief, it may be said, is relevant. (The original application of this kind of idea was to simple inference-drawing computers – if they are not aware of when the body of information input contains a contradiction, you should not allow them to draw conclusions using full classical logic.) The problem with all this is that it doesn't really seem like logic any more. There

are no counterexamples to disjunctive syllogism, that is, structures where $\phi \vee \psi$ and $\neg\phi$ are true but ψ is false. Rather, there are cases where *according to my beliefs* $\phi \vee \psi$ and $\neg\phi$ are true, and also *according to my beliefs* ψ is not true. But it is already well-known that, when we have a non-truth-functional operator like 'according to my beliefs', it won't generally be true that classically valid sequents remain correct when each sentence in the sequent is in the scope of the operator. (Think of the non-truth-functional operator 'possibly': $\phi, \psi \vDash \phi \wedge \psi$ is correct, 'possibly ϕ, possibly $\psi \vDash$ possibly $\phi \wedge \psi$' is not correct – think of the case where ψ = 'the coin toss lands heads', ϕ = 'the coin toss lands tails'.)

16.2 Designators

Designators in $\mathscr{L}_2$ and English The formal properties of $\mathscr{L}_2$ have been established; now we begin a relatively informal investigation into the possible connections between the semantics of $\mathscr{L}_2$ and the semantics of English. We begin in this section with a discussion of *designation* in both languages.

Direct Reference $\mathscr{L}_2$ embodies a particularly simple view of the meaning of a constant.

Definition 114 (Direct Reference). A referring expression *directly refers* iff the meaning of the name is just its referent, the object denoted by the expression.

Constants in $\mathscr{L}_2$ directly refer, because the meaning (semantic value) of a in $\mathscr{A}$ is $[\![a]\!]_{\mathscr{A}}$, which is just an item in the domain of $\mathscr{A}$.

This is to be contrasted with any thesis of *indirect* reference, according to which the semantic value of a denoting expression is something other than the referent. The primary example here are definite descriptions; if Russell's account (recall chapter 13) is correct, for example, the meaning of a description is somehow to be extracted from a complex quantified sentence; the referent (if any) is determined by the meaning, but is not the meaning. This can clearly be seen if we accept Russell's treatment of 'The present king of France is bald', where the sentence is meaningful even though the definite description fails to denote anything. Given that the meaning of the sentence is determined by the meanings of

its constituents, the definite description noun phrase must have a meaning, even though it lacks a referent, and hence cannot be a directly referential expression.

Rigid Designation Direct reference is related to, but distinct from, the following notion:

Definition 115 (Rigid Designator). A referring expression a is a *rigid designator* iff the referent of a is the same object in every possible situation (possible world) in which that object exists.[1]

It may seem that this doesn't hold in $\mathscr{L}_2$, because if $\mathscr{A} \neq \mathscr{B}$, then it may be that $I_{\mathscr{A}}(a) \neq I_{\mathscr{B}}(a)$. But for all that it may hold; the crucial fact to recognise is whether, *in a given situation*, when we consider what is possible for a, we consider what is possible for what a refers to – in every situation in which that thing exists. Since $\mathscr{L}_2$ has no resources for discussing possibility, it is a moot point whether or not rigid designation holds for it.

If an expression is directly referential, in a language with resources for discussing possibility and necessity, then it looks like it will also be a rigid designator. Assume that we evaluate 'Possibly, $F(a)$' by seeing whether there is a possible world in which $F(a)$. If a directly refers, the evaluation of 'Possibly $F(a)$' will proceed by evaluating whether actual meaning of a – which is just some object $\mathbf{a}$, given direct reference – satisfies F at each possibility in which $\mathbf{a}$ exists. So the actual meaning of a, evaluated at each world, will remain constant.

The classic example of a non-rigid designator is also a simple definite description, like 'the tallest person'. Suppose Jacques is the tallest person. Still, I can perfectly well say 'Possibly, Gill is the tallest person', which is *not* synonymous with 'Possibly, Gill is Jacques'. At the possible world in which Gill is the tallest person, the expression 'the tallest person', with its actual meaning, denotes her, though it doesn't denote her actually.

[1] Be careful not to confuse this with the claim that every possible use of 'a' refers to the same thing – this would be false. The idea is rather, when we actually evaluate the possibility of various things about a in some possible world w, we continue to hold fixed the actual meaning of a, rather than (for example) what the residents of w themselves would mean by their own utterance of 'a'. Suppose we consider the possibility in which Aristotle hadn't been called 'Aristotle': we are still talking about the person actually called 'Aristotle', even though residents of that possibility do not use that name to denote anyone.

Things are complicated by the existence of *rigidified descriptions*. As the name suggests, they are rigid designators, but they are also descriptions which refer indirectly. Consider 'the actual tallest person': intuitively, the meaning of this description evaluated at every possibility determines it to have a constant referent, of Jacques. Because of this, 'Possibly, Gill is the actual tallest person' does seem to entail that possibly, Gill is Jacques.

Names and Direct Reference in English For names in English, like 'Antony Eagle' or 'Aristotle', there is no straightforward semantics for names as there is for constants in $\mathscr{L}_2$.

On the one hand, there is considerable syntactic and semantic evidence that names in English function rather like the constants of $\mathscr{L}_2$. Consider

(18) Antony Eagle is lecturing now.

(18) is true iff the thing **Antony Eagle**, designated by 'Antony Eagle', has the property **lecturing** denoted by 'is lecturing', **now**, i.e., at the time designated by (this use of) 'now'. These seem remarkably like the clauses for the satisfaction of atomic formulae of $\mathscr{L}_2$; in that sense it looks like the meaning of 'Antony Eagle' is just the thing it refers to.

Objection to Direct Reference: Informative Identities If 'Hesperus' denotes Hesperus, and 'Phosphorus' denotes Phosphorus, then, since Hesperus *is* Phosphorus (both are names for Venus), then the meanings of 'Hesperus' and 'Phosphorus' are the same. So then why aren't these sentences synonymous?:

(19) Hesperus is Hesperus.

(20) Hesperus is Phosphorus.

This objection trades on the fact that we receive information when told the second claim, but we do not when told the first; hence they must yield different information, and not be synonymous after all. One possible response is to say that *for all we know*, we are in a situation where 'Phosphorus' names something other than Venus; and what we are told by the second sentence is a contingent fact about English, namely, that 'Phosphorus' means Phosphorus!

Objection to Direct Reference: Empty Names If the meaning of a name is its referent, then 'Santa Claus is jolly' is meaningless, yet we judge it to be true. And don't say it is true 'in a fiction' – for fictional English is the same as non-fictional English, so if it is meaningless outside of the fiction, it is just as meaningless inside. But set that aside: *negated existentials* show that 'Santa Claus' is meaningful, because 'Santa Claus does not exist' is straightforwardly true.

To be more explicit: 'Santa Claus does not exist' apparently expresses a truth. This can be rendered in the language $\mathscr{L}_{=}$ by this translation: '$\neg\exists x(x = a)$', where 'a' denotes Santa Claus. But since '$a = a$' expresses a logical truth, it follows that '$\exists x(x = a)$' expresses a logical truth of $\mathscr{L}_{=}$, denying the apparent truth which we translated. But this looks like a problem in English, as well as in formal languages: since 'Santa Claus is identical to Santa Claus' also expresses a truth, the English claim 'Something is identical to Santa Claus' is also true. But if something is identical to Santa Claus, Santa Claus does exist. To deny this is to deny claims that seem tautologous in English.

This is a very serious objection to direct reference theories. Many have chosen to bite the bullet, and accept that such sentences are in fact meaningless. Others have chosen to accept some secondary notion of meaning, like a canonical description associated with the empty name, to play a name-like role in these kind of claims. But there is no consensus on how to deal with this problem.

Objection to Direct Reference: The Role of Descriptions Many names are simply introduced by descriptions; and certainly what most of us aim to communicate when we use a name is a certain individual satisfying a certain canonical description that the hearer is familiar with – if there were no such description, 'how do people ever use names to refer to things at all?' (Kripke, 1980: 28).

Reference Fixing Some descriptions are used to establish the reference: 'Julius' may be introduced into the language as 'the inventor of the zipper'. But it is no part of the meaning: consider the counterfactual situation where Julius gets pipped by his rival.

Background knowledge If one knows a lot about Julius, and one's hearers do too, then one will expect that background knowledge to be useable when one uses the name 'Julius'. It may be difficult if not impossible for a speaker

themselves to separate the meaning of 'Julius' from what they know about Julius.

Opaque Contexts Consider

(21) Lois Lane believes that Superman is Superman;

(22) Lois Lane believes that Superman is Clark Kent.

It is fairly clear that in the context 'Lois Lane believes that Superman is x', one cannot straightforwardly substitute co-referring expressions into x and get a true sentence. Such *opaque contexts* – those which do not permit substitution *salva veritate* – are a problem for the direct reference theory, of course, but also for other theories that think the objects of belief are propositions concerning the entities one has the belief about.

Obviously $\mathscr{L}_2$ lacks opaque contexts, as theorems about substitution of co-referring constants proved in Chapter 10 show. This is an expressive weakness of $\mathscr{L}_2$ compared with English.

Other Designators in English There are several other natural language expressions other than names and definite descriptions which have a designator role in English.

Indexicals such as 'I', 'you', 'here', 'now', 'actually'. This class may also include *demonstratives* like 'this', 'that'.

Count Nouns such as 'chair', 'boy', 'piece of furniture'.

Non-Count Nouns such as 'gold', 'dirt', 'intelligence'.

Count nouns can be handled by the apparatus of quantification in $\mathscr{L}_2$, rather than as designating expressions directly.

Indexicals The characteristic of an indexical expression is *context sensitivity*. 'I' refers to different people when uttered by different people: to me when I say 'I am tired', to you if you utter that same string of words. Similarly with other expressions in this category, as they are sensitive to features of context – where the

context of an utterance includes the speaker, time, place, etc., of the utterance – and thus may have different referents in different contexts.

Despite the context-sensitivity of indexical *reference*, their meaning appears constant. Distinguish (following Kaplan):

Content The content of an expression is its semantic value;

Character The character of an expression is a function from context to content.

The thesis is that indexicals have a constant character, but a variable content; and character is plausibly a kind of meaning.

Pronouns – 'she', 'they', 'it' – may perhaps be analysed similarly.

Non-Count Nouns Non-count nouns are a certain type of *non-plural* referring expression. One reasonable test is whether 'μ' can be grammatically substituted into the expression 'I want some μ'. These fall into at least two kinds:

Mass Mass nouns, like 'gold' or 'dirt', denote a kind of stuff or substance, rather than a particular thing. They typically display *cumulative reference*: if 'being gold' is true of a and b, then it is true of the *sum* $a + b$ which has just a and b as parts.

Abstract Abstract nouns, like 'intelligence' or 'generosity', which seem to denote *qualities* that something may possess, but can themselves have things predicated of them.

A full treatment of these nouns cannot be carried out in $\mathscr{L}_2$; constants referring to qualities, that we naturally translate as predicates like 'being gold', may require a *second order* logic, which permits quantification over predicate variables as well as individual variables.

16.3 More on Relations

Expressing Properties of Binary Relations in $\mathscr{L}_2$ We talked about the properties of binary relations, modelled as sets, back in chapter 2.

We can use formulae of $\mathscr{L}_2$ to express various properties of relations on an entire domain. For example

Reflexivity A relation R is reflexive on $D_{\mathscr{A}}$ if $R = \left[\!\left[P^2\right]\!\right]_{\mathscr{A}}$ and $\left[\!\left[\forall x P^2 x x\right]\!\right]_{\mathscr{A}} = T$.

Transitivity A relation R is transitive on $D_{\mathscr{A}}$ if $R = \left[\!\left[P^2\right]\!\right]_{\mathscr{A}}$ and

$$\left[\!\left[\forall x \forall y \forall z (P^2 x y \wedge P^2 y z) \rightarrow P^2 (x z)\right]\!\right]_{\mathscr{A}} = T.$$

Symmetry A relation R is symmetric on $D_{\mathscr{A}}$ if $R = \left[\!\left[P^2\right]\!\right]_{\mathscr{A}}$ and $\left[\!\left[\forall x \forall y (P^2 x y \rightarrow P^2 y z\right]\!\right]_{\mathscr{A}} = T$.

Relations in English and $\mathscr{L}_2$: Intension and Extension Is the semantic value of an English predicate expression a property or relation as those are interpreted in $\mathscr{L}_2$-structures?

The property 'is a person' on the domain of Oxford students is the same property as 'is a student'; but even though they coincide in their members, we do not think these predicates are synonymous. We may distinguish the property given

- *in extension*, by the list of things which satisfy it; or
- *in intension*, by a rule.

On this domain, two different rules give the same extension; and intuitively in English the meaning is the property given intensionally.

Relations in English and $\mathscr{L}_2$: Sparse and Abundant Properties *Any* set of pairs is a binary relation. Yet this means that 'is identical to' is just as legitimate a relation as 'is within 21 metres of'; whereas we think the former is a more fundamental and non-accidental relation than the latter. We may think that the 'real' properties are sparse and fundamental, and the abundant predicates like 'grue' and 'within 21 metres of' are less so, and do not correspond to genuine properties – but $\mathscr{L}_2$ hardly allows us to draw that distinction.

So despite the fact that 'grue' and 'green' are both meaningful predicates of English, we want to give quite different things to be their semantic values – perhaps just a set of things for 'grue', but a real property **greenness** to be the semantic value of 'green'.

Despite these problems, the case against English having a similar semantics for predicates to $\mathscr{L}_2$ is rather weaker than that against English having a similar semantics for designators.

Further Reading

The best defence of dialethism is Priest (2006); the application of relevant logic to inconsistent data sets is Belnap (1977).

A wonderful book on names and description is Kripke (1980). A modern classic: amazingly, it is more or less a transcription of a series of lectures originally delivered without notes. A useful source on empty names is Caplan (2006). The topic of empty names is one main topic of another famous series of lectures by Kripke (2013).

Exercises

1. (a) Show that the following derivation rules are acceptable in our framework:
   i. *Modus Ponens*: If $\vDash \phi$ and $\vDash \phi \to \psi$, then $\vDash \psi$.
   ii. *Disjunctive Syllogism*: If $\vDash \phi \vee \psi$ and $\vDash \neg\phi$, then $\vDash \psi$.
   iii. *Reductio ad Absurdum*: If $\phi \vDash \psi$ and $\phi \vDash \neg\psi$ then $\vDash \neg\phi$.

   (b) In our system each acceptable rule of inference has a corresponding correct semantic sequent (for example, the sequent $\phi, (\phi \to \psi) \vDash \psi$ corresponds to *modus ponens*). Consider now this rule of inference in English: if ϕ is a truth of logic, then 'Necessarily, ϕ' is too. Is this rule intuitively correct? Is the corresponding sequent '$\phi \vDash_{\text{English}}$ Necessarily ϕ' intuitively correct?

2. Say that $\phi \vDash \psi$ is a *perfect* sequent iff it is correct, and ϕ is satisfiable, and ψ is not a tautology. Say that a sequent is *perfectible* iff it is a substitution instance of a perfect sequent.

   (a) Show that there must be a sentence letter in common between ϕ and ψ is a perfectible sequent. (Hint: use the Craig Interpolation Theorem (page 88), plus properties of substitution.)

   (b) Give examples to show that perfectible entailment is not transitive (i.e., that there are cases where ϕ perfectibly entails ψ, where ψ perfectibly entails χ, but where ϕ does not perfectibly entail χ.)

   (c) Which of the structural rules of §4.8 (page 66) are perfectibly inappropriate (i.e., take perfectible sequents to non-perfectible sequents.)?

3. Which of the relations expressed by the following English predicates are equivalence relations:

   (a) 'x entails y', on the domain of sentences of $\mathscr{L}_2$.

   (b) 'x is logically equivalent to y', on the domain of sentences of $\mathscr{L}_2$.

   (c) 'If $\{x\}$ is consistent, then $\{x\} \cup \{y\}$ is consistent' on the domain of sentences of $\mathscr{L}_2$.

4. Show that $\forall x \forall y(\neg x = y \rightarrow (\boldsymbol{P}xy \vee \boldsymbol{P}yx) \equiv \forall x \forall y(x = y \vee \boldsymbol{P}xy \vee \boldsymbol{P}yx)$.

5. Show by suitable reasoning that in a finite domain, for any *partial order* $\boldsymbol{R}$, $\exists x \forall y(\boldsymbol{R}yx \rightarrow x = y)$. Give a counterexample to this condition in an infinite domain.

6. Give a graph, on some non-empty domain, of a relation $\boldsymbol{R}$ which satisfies this condition: $\forall x \forall y(\boldsymbol{R}xy \rightarrow \exists z((x \neq z) \wedge (y \neq z) \wedge \boldsymbol{R}xz))$. Can you give an example of a relation which satisfies this condition (being sure to specify the domain)?

7. We might normally expect 'is similar to' to be a symmetric relation: after all, if there is a respect in which a is similar to b, then b must be similar to a in that very same respect. But many people seem to judge that similarity is *not* symmmetric:

   > When people are asked to make comparisons between a highly familiar object and a less familiar one, their responses reveal a systematic asymmetry: The unfamiliar object is judged as more similar to the familiar one than vice versa. For example, people who know more about the USA than about Mexico judge Mexico to be more similar to the USA than the USA is to Mexico. (Kunda, 1999: 520)

   (You might think also of the fact that it is much more natural to say that children resemble their parents, than that parents resemble their children.)

   Can you provide a rationale for these psychological results? Do they indicate that people are systematically mistaken about the meaning of the relational predicate 'is similar to', or do they indicate that our theory of similarity in terms of matching respects of similarity is incorrect?

Appendix A

Answers to Selected Exercises

Exercises for Chapter 1: page 14

1. We want to show that the LNP holds. The LNP is a conditional claim: so we'll assume the antecedent ('left hand side') of the conditional, and show that the consequent ('right hand side') must then hold, by reasoning using the strong principle.

   We'll reason by *reductio*. So we'll assume that there is a non-empty set of natural numbers M with *no* least member. So now consider this property of natural numbers: *n is not being a member of* M, symbolised $n \notin M$.

   Pick some number k at random. Suppose that none of k's predecessors are in M. Could k be in M? If it were, it would be the least member of M, because it is the earliest member of the natural number sequence to be found in M. But M has no least member. So $k \notin M$. Discharge our supposition, we get this conditional: *if for all* $n < k$, $n \notin M$, *then* $k \notin M$. Since k was random, we didn't use any specific features of k at all. So this reasoning would hold for any number at all. So we have this claim: *For all k (if for all* $n < k$, $n \notin M$, *then* $k \notin M$*)*. This is in the right form to apply strong induction. So we conclude: *for all k,* $k \notin M$.

   But now we have a problem: for if no number is a member of M, and M is a subset of the natural numbers, then there is only one option: M is empty. Our supposition was that M is non-empty. So collectively, our suppositions have led to a contradiction. We blame the supposition that M has no least member, so we conclude that if M is a non-empty set of natural numbers, then it must have some least member.

2. Suppose there is some sentence operator *not* that when applied to some sentence

P yields a sentence *not-P* which is true just in case P is false, and false just in case P is true. We can use this principle, along with plausible principles about truth and disjunction, to argue that Bivalence leads to the LEM.

If Bivalence is true, then for every meaningful sentence P, either P is true or P is false. By our assumption about *not*, it follows that either P is true or not-P is true. But then 'P or not-P' is true (by the principle that if ϕ is true or ψ is true, then 'ϕ or ψ' is true). If ϕ is true, then ϕ. So for any meaningful P, P or not-P, which is the LEM.

The argument from LEM to Bivalence is trickier. We cannot simply run the above argument in reverse. Suppose we try: we assume the LEM, and then conclude that for each P, 'P or not-P' is true. Does this mean that either P is true or 'not-P' is true?

Not necessarily. Consider the case of vagueness. Suppose Alice is borderline tall, neither definitely tall nor definitely not tall. It is natural to say that a sentence can be true only if it is definite. A sentence is definite if it doesn't contain any vague terms, or if it does contain them, they don't make a difference. One influential approach to vagueness known as *supervaluationism* (Fine, 1975) says that the vague terms don't make a difference just in case the sentence would be true no matter how the vague terms are made precise. So the sentence 'Alice is tall' is indefinite, because it contains a vague term and on some ways of making 'tall' precise, it is true – those where we make it precise by setting the boundary between tall and not-tall below Alice's height - and on other ways of making it precise, it is false. So 'Alice is tall' is indefinite, and thus *neither true nor false*. Hence Bivalence fails, according to the supervaluationist.

But what about the sentence 'Alice is tall or not-(Alice is tall)'? No matter where we draw the boundary for 'tall' – so long as we do it the same way in each disjunct – Alice will fall on one side or the other. So that sentence is definite: no matter how we make the vague expression precise, it doesn't change the truth value. So the whole sentence is definite. In fact, every instance of LEM is definitely true according to the supervaluationist. So while 'P or not-P' is always true, that doesn't entail – says the supervaluationist – that either P is true or 'not-P' is true. So the supervaluationist, at least, wants to resist the argument from LEM to Bivalence.

Exercises for Chapter 2: page 39

1. (a) $X \subseteq Y$ iff each member of X is a member of Y. $Y \subseteq X$ iff each member of Y is a member of X. If both hold, then X and Y have the same members. By extensionality, $x = Y$.

(b) If $X \subseteq Y$ then every member of X is also a member of Y. If $Y \subset Z$, all the members of Y (which include among them all the members of X) are members of Z. So all the members of X are also members of Z, i.e., $X \subseteq Z$.

(d) If $X \subset Y$, then all members of X are members of Y, but not vice versa. So there is at least one member of Y which is not a member of X. So $Y \not\subset X$.

2. (a) $X \cap X$ is the set which has all the members in both X and itself; but that is just the set which shares all its members with X, namely X (by extensionality). Similar reasoning shows $X \cup X = X$.

(c) The set which contains all the members of X and all the members of the empty set can be distinct from X only if there are members of the empty set, which there are not.

(e) $X \cap (Y \cup Z)$ is the set containing just those members of X which are also members of Y or members of Z. There are thus two ways to be a member of this set: be a member of X and Y, or be a member of X and Z (or both). That is equivalent to: be a member of $X \cap Y$ or a member of $X \cap Z$ (or both). That is equivalent to: be a member of $(X \cap Y) \cup (X \cap Z)$. Here we have exploited the connection between conjunction and intersection, and disjunction and union, mentioned in the text.

(g) The set $X \setminus (Y \cap Z)$ is the set of those members of X which are not members of both Y and Z. Something can be a member of this set just in case it is in X but not in Y, or is in X but not in Z (or both). Equivalently: be a member of $X \setminus Y$ or a member of $X \setminus Z$; equivalently, be a member of $(X \setminus Y) \cup (X \setminus Z)$.

4. (a) $\langle\!\langle x, y \rangle\!\rangle = \langle\!\langle u, v \rangle\!\rangle$ iff $\{\{x\}, \{x, y\}\} = \{\{u\}, \{u, v\}\}$ iff either (i) $\{x\} = \{u\}$ and $\{x, y\} = \{u, v\}$ or (ii) $\{x\} = \{u, v\}$ and $\{x, y\} = \{u\}$. Case (ii) only holds when $x = y = u = v$, because extensionality entails that identical sets must have the same number of members. In that degenerate case, the theorem holds. In the non-degenerate case where $x \neq y$, case (i) must hold. But case (i) holds iff $x = u$ and $y = v$, by extensionality again.

(b) If $x = y$ then $\langle\!\langle x, y \rangle\!\rangle = \{\{x\}, \{x, x\}\} = \{\{x\}, \{x\}\} = \{\{x\}\}$.

5. (c) The powerset of any set Z contains sets which have members of Z as their only members. So $\wp(X)$ only contain sets of things from X; $\wp(Y)$ only contains sets of things from Y. So all the members of $\wp(X) \cup \wp(Y)$ are 'pure' sets – it has no members which are sets mixing members of X and Y (except if $X \cap Y$ is non-empty). By contrast, $\wp(X \cup Y)$ does contain such mixed sets. In fact, it is easy to see that as long as there is an $X \setminus Y$ and $Y \setminus X$ are both non-empty, then $\wp(X) \cup \wp(Y) \subset \wp(X \cup Y)$.

   For example, let $X = \{1, 2\}, Y = \{2, 3\}$. $\{1, 3\} \in \wp(X \cup Y)$, but $\{1, 3\} \notin \wp(X) \cup \wp(Y)$; by extensionality, $\wp(X) \cup \wp(Y) \neq \wp(X \cup Y)$.

17. (c) i. The successor function which maps each number to its immediate successor (i.e., $\mathsf{succ}(x) = x + 1$) is into $\mathbb{N}$ but not onto, since there is no n such that $\mathsf{succ}(n) = 0$ but $0 \in \mathbb{N}$.

   ii. The identity function from natural numbers into integers $\mathbb{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3\ldots$ is one-one, since each number is mapped to itself and no other number. But it is not a bijection, because there is no natural number n such that $n = -1$.

   iii. Any function which is not one-one has no inverse. So consider the squaring function on integers, $\mathsf{sq}(x) = x^2$ for any $x \in \mathbb{Z}$. Since $\mathsf{sq}(2) = \mathsf{sq}(-2) = 4$, the relation which is the inverse of sq contains the pairs $\langle 4, 2\rangle$ and $\langle 4, -2\rangle$, and so is not a function.

   iv. An argument which is its own inverse is such that $f(f(x)) = f^{-1}(f(x)) = x$. Identity is its own inverse. A more interesting case is negation n, discussed in chapter chapter 4. $\mathsf{n} = \{\langle T, F\rangle, \langle F, T\rangle\}$; swap the order of each pair, and we get the same set back. Another example is the function rotate 180° on the domain of compass directions: apply the function twice, and we get the original argument back again.

   (d) The union of two functions is itself a function iff the constituent functions are not defined on any argument(s) in common. Since both constituent functions are relations, their union is a relation too (though perhaps on a different – larger – domain); and that relation meets the condition for being a function iff any a which is in the domain of one constituent function and in the domain of the other is such that both functions assign to a that same value.

Exercises for Chapter 3: page 51

1. Answer:

For any numbers n and m, $\ulcorner n + m \urcorner$ denotes a number, as long as '+' denotes the addition operator.

3. Suppose that ϕ is a non-atomic $\mathscr{L}_1$ sentence which is formed in accordance with two formation rules. There are two possible cases: (i) ϕ begins with $\neg$. If ϕ was also produced by one of the other formation rules, then $\phi = \ulcorner(\chi \oplus \pi)\urcorner$. If so, '$\neg$' would have to be the same symbol as '(', which contradicts our supposition that no connective is also a piece of punctuation. (ii) $\phi = \ulcorner(\chi \oplus \pi)\urcorner$ and $\phi = \ulcorner(\psi \otimes \omega)\urcorner$ for sentences χ, π, ψ, ω and binary connectives $\oplus, \otimes$. There are three options for where the relevant occurence of $\otimes$ can occur in $\ulcorner(\chi \oplus \pi)\urcorner$: it can occur somewhere in χ, it can occur at $\oplus$, or it can occur somewhere in π.

   - Suppose it occurs in χ. Then $\chi = \ulcorner\psi \otimes \sigma\urcorner$ for some string σ. By Theorem 20, since χ is a sentence, ψ is not, contradiction.
   - So suppose it occurs in π. Then $\psi = \ulcorner\chi \oplus \sigma'\urcorner$ for some string σ'. Again by Theorem 20, since ψ is a sentence, χ is not, contradiction.
   - So $\ulcorner\oplus = \otimes\urcorner$. But that would contradict our intitial assumption that 'no character in any category is identical to any other character in that category'.

   So there couldn't be any such sentence ϕ.

4. Suppose ϕ is any $\mathscr{L}_1$ sentence, and every constituent sentence of ϕ of lower complexity is uniquely readable. ϕ was either formed by applying negation to a uniquely readable sentence, or applying one of the binary connectives to uniquely readable sentences. By the result of the previous exercise, there is exactly one formation rule that could have been applied to produce ϕ, and it is uniquely determined which formation rule was applied to construct ϕ. So ϕ too is uniquely readable. By strong induction, every sentence of $\mathscr{L}_1$ is uniquely readable.

Exercises for Chapter 4: page 76

1. $\Gamma \vDash \phi$ iff every $\mathscr{L}_1$-structure $\mathscr{A}$ in which each member of Γ is true is also one in which ϕ is true; iff every $\mathscr{L}_1$-structure $\mathscr{A}$ in which each member of Γ is true is also one in which $\neg\phi$ is false; iff every $\mathscr{L}_1$-structure $\mathscr{A}$ is one in which at least one member of $\Gamma \cup \{\neg\phi\}$ is false; iff $\Gamma \cup \{\neg\phi\}$ is unsatisfiable.

2. (a) When $\Gamma \vDash \phi$, every line of the truth table on which each member of Γ gets a T will be one on which ϕ also gets a T.

(b) When Γ is consistent, at least one line of the truth table will assign a T to each member of Γ.

(c) When ϕ is a contradiction, it will get F on every line of the truth table.

(d) When ϕ and ψ are logically equivalent, they get the same value on every line of the truth table.

4. (a) Suppose $\chi = (\phi \wedge \psi)$. Then $[\![\chi]\!]_{\mathscr{A}} = F$ iff either $[\![\phi]\!]_{\mathscr{A}} = F$ or $[\![\psi]\!]_{\mathscr{A}} = F$ iff either $[\![\phi]\!]_{\mathscr{A}} \neq T$ or $[\![\psi]\!]_{\mathscr{A}} \neq T$ iff $[\![\phi \wedge \psi]\!]_{\mathscr{A}} \neq T$ iff $[\![\chi]\!]_{\mathscr{A}} \neq T$.

(b) Suppose $\chi = (\phi \rightarrow \psi)$. Then $[\![\chi]\!]_{\mathscr{A}} = F$ iff $[\![\phi]\!]_{\mathscr{A}} = T$ and $[\![\psi]\!]_{\mathscr{A}} = F$ iff $[\![\phi]\!]_{\mathscr{A}} \neq F$ and $[\![\psi]\!]_{\mathscr{A}} \neq T$ iff $[\![\phi \rightarrow \psi]\!]_{\mathscr{A}} \neq T$ iff $[\![\chi]\!]_{\mathscr{A}} \neq T$.

(c) Suppose $\chi = (\phi \leftrightarrow \psi)$. Then $[\![\chi]\!]_{\mathscr{A}} = F$ iff $[\![\phi]\!]_{\mathscr{A}} \neq [\![\psi]\!]_{\mathscr{A}}$ iff $[\![(\phi \leftrightarrow \psi)]\!]_{\mathscr{A}} \neq T$ iff $[\![\chi]\!]_{\mathscr{A}} \neq T$.

9. If ϕ and ψ express the same truth-function f_n, and they contain the same sentence letters $s_1, \ldots, s_n$ then in every structure $\mathscr{A}$,

$$f(\mathscr{A}(s_1), \ldots, \mathscr{A}(s_n)) = [\![\phi]\!]_{\mathscr{A}} = [\![\psi]\!]_{\mathscr{A}}\,.$$

But then these two sentence have the same truth value in every $\mathscr{L}_1$-structure, and are logically equivalent. The restriction to sentences with the same sentence letters is crucial. $(P \wedge Q)$ and $(P \wedge R)$ both express the conjunction function $\mathbf{k}$, but are not logically equivalent.

If ϕ and ψ contain the same n sentence letters and are logically equivalent, then in every $\mathscr{L}_1$-structure $\mathscr{A}$, $[\![\phi]\!]_{\mathscr{A}} = [\![\psi]\!]_{\mathscr{A}}$. If ϕ expresses f, then $f(\mathscr{A}(s_1), \ldots, \mathscr{A}(s_n)) = [\![\phi]\!]_{\mathscr{A}} = [\![\psi]\!]_{\mathscr{A}}$. But since ψ contains just $s_1, \ldots, s_n$ too, ψ also expresses f.

10. The underlying result is this: for any structure $\mathscr{A}$,

$$[\![\neg(\phi \leftrightarrow \psi)]\!]_{\mathscr{A}} = [\![(\neg\phi \leftrightarrow \psi)]\!]_{\mathscr{A}}\,.$$

For $[\![\neg(\phi \leftrightarrow \psi)]\!]_{\mathscr{A}} = T$ iff $[\![\phi \leftrightarrow \psi]\!]_{\mathscr{A}} = F$ iff $[\![\phi]\!]_{\mathscr{A}} \neq [\![\psi]\!]_{\mathscr{A}}$ iff $[\![\neg\phi]\!]_{\mathscr{A}} = [\![\psi]\!]_{\mathscr{A}}$ iff $[\![\neg\phi \leftrightarrow \psi]\!]_{\mathscr{A}} = T$.

Suppose we have some sentence of our language $\mathscr{L}_{\leftrightarrow,\neg}$, which has a constituent which is a negated biconditional. We can apply the above result, and Theorem 46, to derive that the sentence obtained by replacing this constituent by 'pushing in' the negation to apply only to the antecedent of the biconditional is logically equivalent to our original sentence. If we apply this procedure again to the new sentence, and repeatedly the sentence obtained as a result of the procedure, we will eventually

drive all negations to rest against sentence letters, and none will have any occurrence of $\leftrightarrow$ in their scope. The procedure always reduces the complexity of any negated biconditional sentence, so it will terminate eventually because every sentence has finite complexity. (In effect we have an induction on the maximal complexity of constituents which are negated biconditionals: the procedure shows that if we have a sentence with no negated biconditional constituent of complexity greater than i, that is logically equivalent to a sentence with containing no negated biconditional constituent of complexity greater than $i-1$; if our induction hypothesis is that all sentences with a negated biconditional constituent of complexity no greater than $i-1$ are equivalent to sentences with no biconditional in the scope of negation, we have our result. The procedure described may be a little easier to intuitively grasp than induction on complexity of constituents.)

So consider $\neg(\neg(P \leftrightarrow Q) \leftrightarrow R)$. We see that the sentence is a negated biconditional, so we apply the procedure to obtain $(\neg\neg(P \leftrightarrow Q) \leftrightarrow R)$. We apply the procedure to $\neg(P \leftrightarrow Q)$ obtaining $(\neg(\neg P \leftrightarrow Q) \leftrightarrow R)$. Finally, we apply the procedure to $\neg(\neg P \leftrightarrow Q)$, obtaining $((\neg\neg P \leftrightarrow Q) \leftrightarrow R)$.

Exercises for Chapter 5: page 101

1. (b) $[\![\phi \wedge \psi]\!]_{\mathscr{A}} = \max\left([\![\phi]\!]_{\mathscr{A}}, [\![\psi]\!]_{\mathscr{A}}\right) = \max\left([\![\psi]\!]_{\mathscr{A}}, [\![\phi]\!]_{\mathscr{A}}\right) = [\![\psi \wedge \phi]\!]_{\mathscr{A}}$, by commutatively of 'max'.

   (c) $\mathbf{k}$ is commutative and associative. For, e.g., $\mathbf{k}\left([\![\phi]\!]_{\mathscr{A}}, [\![\psi]\!]_{\mathscr{A}}\right) = [\![\phi \wedge \psi]\!]_{\mathscr{A}} = [\![\psi \wedge \phi]\!]_{\mathscr{A}} = \mathbf{k}\left([\![\psi]\!]_{\mathscr{A}}, [\![\phi]\!]_{\mathscr{A}}\right)$. Similarly, $\mathbf{k}(x, \mathbf{k}(y, z)) = \mathbf{k}(\mathbf{k}(x, y), z)$.

4. Our task is to construct a sentence in conjunctive normal form that expresses any arbitrary truth function. Suppose we are given an n-place truth function. Select n sentence letters $\{s_1, \ldots, s_n\}$, and construct a sentence using them as follows.

   - If in some structure $\mathscr{A}$, $f([\![s_1]\!]_{\mathscr{A}}, \ldots, [\![s_n]\!]_{\mathscr{A}}) = F$, call $\mathscr{A}$ an f-disagreeing structure. If $\mathscr{A}$ is an f-disagreeing structure, let

     $$S_i^{\mathscr{A}} = \begin{cases} s_i & \text{if } [\![s_i]\!]_{\mathscr{A}} = F \\ \neg s_i & \text{if } [\![s_i]\!]_{\mathscr{A}} = T. \end{cases}$$

     For each f-disagreeing structure $\mathscr{A}$, we can see that $\left[\!\left[S_i^{\mathscr{A}}\right]\!\right]_{\mathscr{A}} = F$.

   - Define $\mathfrak{d}_{\mathscr{A}} = \left[\bigvee_{i=1}^{n} S_i^{\mathscr{A}}\right]$ for each f-disgreeing structure $\mathscr{A}$. Since in each f-disagreeing structure, the disjuncts of this disjunction are all F, the disjunction $\mathfrak{d}_{\mathscr{A}}$ also is F in an f-disagreeing structure $\mathscr{A}$. Indeed, by construction,

$\mathfrak{d}_{\mathscr{A}}$ is false only in $\mathscr{A}$ and structures that agree with $\mathscr{A}$ on sentence letters in $\{s_1, \ldots, s_n\}$ – only those structures are guaranteed to be ones where every disjunct is false.

There are only finitely many equivalence classes of structures that agree on the setence letters in $\{s_1, \ldots, s_n\}$; for each such class which contains an f-disagreeing structure, pick some f-disgreeing structure from it at random. There are finitely many of these chosen f-disagreeing structures which can be enumerated $\mathscr{A}_1, \ldots \mathscr{A}_m$.

- Define $\mathfrak{c} = \left[\bigvee_{j=1}^{m} \mathfrak{d}_{\mathscr{A}_j}\right]$. This conjunction is false whenever it has any false conjunct, and true otherwise. It is thus false on every f-disagreeing structure, and true otherwise. So it expresses f.

  If there are no f-disagreeing structures, let $\mathfrak{c} = (P \vee \neg P)$, which is obviously true on every structure, just like f, and is degenerately in CNF.

What we have done is create a sentence which says, 'we are not in row x *and* we are not in row y *and* ... *and* we are not in row z', for each row of the truth table for $s_1, \ldots, s_n$ where f gets the value F.

5. Recall that a truth-function is positive iff $f(T, \ldots, T) = T$. Suppose ϕ expresses f. We want to show that if ϕ contains only $\rightarrow$ and $\wedge$ as its connectives, f is positive.

   Base case. ϕ is a sentence letter, s. If s expresses f, then $f([\![s]\!]_{\mathscr{A}}) = \mathscr{A}(s)$. So if $\mathscr{A}(s) = T$, $f(T) = T$. So f is positive.

   Induction step. ϕ is complex (a conjunction or conditional), and the induction hypothesis holds of its simpler constituents ϕ and ψ. Then ψ and χ both express positive truth functions, f_ψ and f_χ – so $f_\psi(T, \ldots T) = T = f_\chi(T, \ldots, T)$.

   - $\phi = (\psi \wedge \chi)$. ϕ then expresses the truth function $\mathbf{k}(f_\psi(\cdots), f_\chi(\cdots))$. Since those constituent truth functions are positive,

     $$\mathbf{k}(f_\psi(T, \ldots, T), f_\chi(T, \ldots, T)) = \mathbf{k}(T, T) = T.$$

     So ϕ expresses a positive truth function too.

   - $\phi = (\psi \rightarrow \chi)$. ϕ then expresses the truth function $\mathbf{c}(f_\psi(\cdots), f_\chi(\cdots))$. Since those constituent truth functions are positive,

     $$\mathbf{c}(f_\psi(T, \ldots, T), f_\chi(T, \ldots, T)) = \mathbf{c}(T, T) = T.$$

     So ϕ expresses a positive truth function too.

6. (b) Yes; $(\neg\phi \rightarrow \psi)$ is logically equivalent to the disjunction $(\phi \vee \psi)$, and we already know that disjunction and negation are together expressively adequate.

(d) No. The two-place connectives are all positive, and so cannot ever express a truth-function which is not positive. Negation is not positive, but no sentence involving only negation can be used to express any constant one-place truth function, since $\mathbf{n}(x) \neq x$, but for a constrant truth function $f(T) = T$ or $f(F) = F$.

(e) $\downarrow$ is expressively complete: $(\phi \downarrow \phi)$ is logically equivalent to $\neg\phi$; and $((\phi \downarrow \psi) \downarrow (\phi \downarrow \psi))$ is logically equivalent to $(\phi \vee \psi)$. And we know that $\{\neg, \vee\}$ is expressively complete.

(f) A 2-place truth-function is expressively adequate if it is able to express all 16 2-place truth functions.

- It cannot be positive (else it would not be able to express those 8 truth-functions which get the value F when given T, T as input) - this rules out the 8 positive truth functions, leaving 8.
- It cannot be negative (i.e., $f(F, F) = F$), since it would not be able to express those 8 truth functions which are not negative.
- This leaves 4 candidates:

		$f_{\uparrow}$	$f_{\text{n-right}}$	$f_{\text{n-left}}$	$f_{\downarrow}$
T	T	F	F	F	F
T	F	T	T	F	F
F	T	T	F	T	F
F	F	T	T	T	T

But $f_{\text{n-left}}$ is the truth-function $f(x, y) = \mathbf{n}(x)$, and $f_{\text{n-right}}$ is the truth-function $f(x, y) = \mathbf{n}(y)$. Neither of these truth-functions is able to express any of the 8 truth functions where $f(F, T) = f(T, F)$.

So only the connectives which express $f_{\uparrow}$ and $f_{\downarrow}$ are expressively complete by themselves.

(g) $\bot$ is not expressively adequate by itself, since it is not true in any structure and so cannot express any positive truth-function. However, $\{\rightarrow, \bot\}$ is expressively adequate: $(P \rightarrow \bot)$ expresses the same truth-function as $\neg P$ (because that conditional expresses the function that yields F iff only if P is true and $\bot$ is false, but since $\bot$ is always false, iff P is true). And it can be easily verified that $((P \rightarrow \bot) \rightarrow Q)$ expresses disjunction $(P \vee Q)$. And conjunction isn't expressively adequate itself.

7. (a) We can express negation using $\rightarrow$ and $\rightarrow^\star$ as follows: $P \rightarrow^\star (P \rightarrow P)$ (or $P \rightarrow (P \rightarrow^\star P)$). Since $\{\neg, \rightarrow\}$ are expressively adequate, so too is $\{\rightarrow, \rightarrow^\star\}$.

Self-duality is not sufficient for expressive adequacy, since $\{\neg\}$ is a self-dual set without being expressively adequate. Nor is it necessary, since $\{\uparrow\}$ is expressively adequate without being a self-dual set.

(b) The definition of duality is such that if $f(x, y) = z$ then $f^\star(x^\star, y^\star) = z^\star$. If $f = f^\star$, so that f is self-dual, then if $f(T, F) = z$, $f(F, T) = z^\star$, so $f(T, F) \neq f(F, T)$. Likewise $f(T, T) \neq f(F, F)$.

There are 16 2-place truth-functions. But 12 of them are such that either the top line or bottom line of the truth-table are the same, or the two middle lines of the truth table are the same. So only 4 could be self-dual: the four degenerate truth-functions that track their first input or second input or their negations, $f_{\text{left}}, f_{\text{n-left}}, f_{\text{right}}, f_{\text{n-right}}$. These are self-dual, basically because when we 'flip' the truth table, these functions directly follow the flipped inputs to the truth-functions, undistorted by any contribution from the other input.

9. (a) $(\neg Q \vee R)$;

(c) $(P \rightarrow Q)$.

10. (b) If $\Gamma \models^\star \Delta$, then every structure which satisfies all of Γ satisfies at least one of Δ. So there can be no structure which satisfies all of Γ and none of Δ. Let $\widetilde{\Delta}$ be the set which results from taking every member of Δ and negating it (i.e., $\widetilde{\Delta} = \{\neg\delta : \delta \in \Delta\}$). Then $\Gamma \cup \widetilde{\Delta}$ is unsatisfiable. By Compactness, some finite subset of $\Gamma \cup \widetilde{\Delta}$ is unsatisfiable – call that set $G\widetilde{D}$. There are three cases to consider:

- $G\widetilde{D}$ contains elements from both Γ and $\widetilde{\Delta}$. Let Φ be the conjunction of all sentences in both Γ and $G\widetilde{D}$. Since the latter is finite, the conjunction is finite: $(\phi_1 \wedge \ldots \wedge \phi_m)$. Let Ψ' be the conjunction of all sentences in $\widetilde{\Delta}$ and $G\widetilde{D}$. Since Ψ' is a conjunction of negated sentences $(\neg\psi_1 \wedge \ldots \wedge \neg\psi_n)$, it is logically equivalent to a negated disjunction $\neg(\psi_i \vee \ldots \vee \psi_n) = \neg\Psi$. (Note that Ψ is the embedded disjunction.) Since $G\widetilde{D}$ is unsatisfiable, $\Phi, \neg\Psi \models$; by Theorem 26, $\Phi \models \Psi$, and Φ and Ψ are of the right form.
- $G\widetilde{D}$ contains elements only from Γ. Let Φ be the conjunction of every sentence in $G\widetilde{D}$, and let Ψ be an arbitrary finite disjunction of members from Δ. Since $G\widetilde{D} \subseteq \Gamma$, Φ is of the right form, and since Φ is a contradiction, $\Phi \models \Psi$.

- $G\widetilde{D}$ contains elements only from $\widetilde{\Delta}$. Then every structure makes some member of Δ true whose negation is in $G\widetilde{D}$. Let Ψ be the disjunction of negations of every sentence in $G\widetilde{D}$; this is a finite tautologous disjunction. Let Φ be an arbitrary finite conjunction of members from Γ. Since Ψ is a tautology, $\Phi \vDash \Psi$.

11. (a) The set of effective procedures is roughly the set of finite recipes in English. Each such recipe is a finite sequence of characters drawn from a finite alphabet, and by the same argument as in the Key Lemma 22, there are only countably many effective procedures. We suppose they can be recursively enumerated because the set of finite sets of English sentences can be effectively generated – this follows from the fact that we can effectively generate finite subsets of $\mathbb{N}$.

(b) i. If there were an effective procedure for computing d, it would be a member of E – say it would be the k-th recipe under the relevant enumeration of recipes. But if d is computed by the k-th recipe, then $d = f_k$. But $d(k) \neq f_k(k)$, by construction of d. So d cannot be computed by the k-th recipe. Since k was arbitrary, d cannot be computed by any effective recipe specifiable in English.

ii. Suppose there were an effective procedure for sorting recipes into those which are *good* (compute a function in a finite time) and those which are *garbage* (not computing a function, not halting in a finite time.) This procedure would determine a function:

$$g(i) = \begin{cases} 1 & \text{iff the } i\text{-th recipe in } E \text{ is good;} \\ 0 & \text{otherwise.} \end{cases}$$

If this function g is effectively computable, then d is effectively computable. The recipe for computing each $d(n)$ is this:

```
On given input n, compute g(n)
    If g(n) = 0, then set d(n) = 1.
    If g(n) = 1, then compute f_n(n)
        If f_n(n) = 1, then set d(n) = 2;
        If f_n(n) ≠ 1, then set d(n) = 1.
```

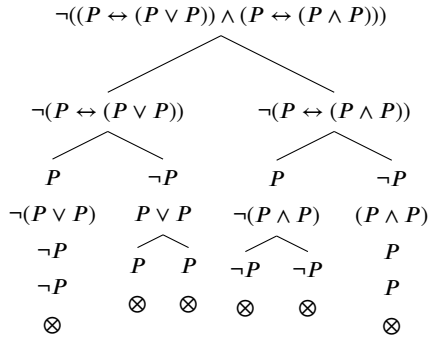
If g is a good test for effective procedures, every function in this little flowchat is calculate by an effective procedure, so for each n, we determine the value of $d(n)$ in a finite time. So d is effectively computable. But

the previous exercise just showed that d is not effectively computable, hence g is not effectively computable.
This shows that the set of effective procedures is recursively enumerable but not recursive.

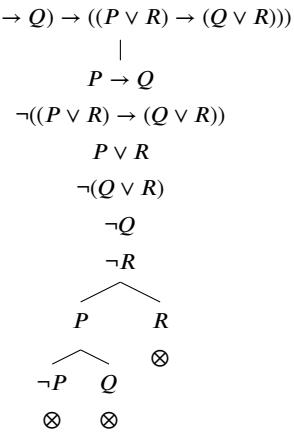
Exercises for Chapter 6: page 128

1. $\neg\neg(P \vee Q)$
$\neg P$
$\neg Q$
$P \vee Q$
$P \quad Q$
$\otimes \quad \otimes$

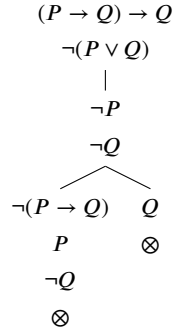
2. (a)



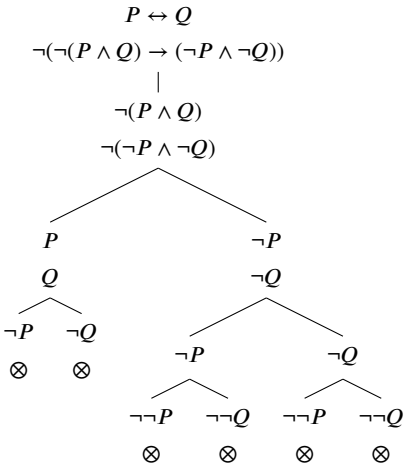
(e)



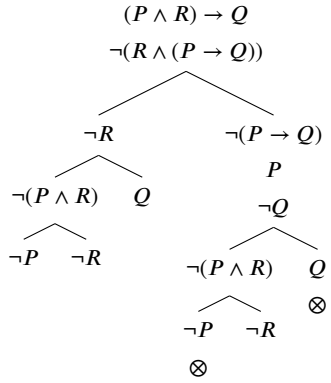
3. (d)


(P → Q) → Q
¬(P ∨ Q)
¬P
¬Q
¬(P → Q)
Q
P
⊗
¬Q
⊗

(e)


P ↔ Q
¬(¬(P ∧ Q) → (¬P ∧ ¬Q))
¬(P ∧ Q)
¬(¬P ∧ ¬Q)
P
Q
¬P
¬Q
¬P
¬Q
¬P
¬Q
⊗
⊗
¬¬P
¬¬Q
¬¬P
¬¬Q
⊗
⊗
⊗
⊗

4. (a)


(P ∧ R) → Q
¬(R ∧ (P → Q))
¬R
¬(P → Q)
¬(P ∧ R)
Q
P
¬Q
¬P
¬R
¬(P ∧ R)
Q
⊗
¬P
¬R
⊗

(b)
$$\begin{array}{c} P \vee Q \\ P \\ \neg\neg Q \\ | \\ Q \\ \diagup\ \diagdown \\ P \quad Q \end{array}$$

6. Any finished tableau generated by $\{\neg\phi\}$ is closed, because that is what $\vdash \phi$ means. Consider a tableau T generated by $\neg(\psi \rightarrow \phi)$. We apply the negated conditional rule, adding ψ and $\neg\phi$ below the root node. Then we extend the resulting tableau by adding the subsequent contents of every branch on some finished tableau generated by $\{\neg\phi\}$ to every branch stemming from $\neg\phi$ on $\mathbf{T}$. So $\mathbf{T}$ is closed (it may not be finished, if there are tableau rules to apply to ψ; but it doesn't matter, since adding more sentences cannot unclose a closed branch). So $\vdash \psi \rightarrow \phi$.

Exercises for Chapter 7: page 149

1.

2. (a) The truth function **nor**. Here is the informal argument. In any tableaux system, for each connective, there is a rule for that connective, and a rule for the negated connective. The left displayed rule clearly concerns the connective $\odot$; so perhaps the right displayed rule concern its 'negation'? If so, given that right-hand rule is of the form $\chi \odot \chi$, *that* must be how to express negation in this system. So the left-hand rule now says: from $\phi \odot \psi$, conclude both that ϕ is false and ψ is false; so $\phi \odot \psi$ is sufficient for neither of ϕ or ψ to be true. The right-hand rule says: from the falsity of $\phi \odot \psi$, conclude that either ϕ or ψ is true. From both rules together, therefore, we know that both ϕ and ψ being false is necessary and sufficient for $\phi \odot \psi$ to be true, i.e., that $\odot$ expresses the truth function $\mathbf{nor}(x, y) = 1$ iff $x = 0$ and $y = 0$.

   (b) If both ϕ and $\phi \odot \phi$ occur on a branch.

Exercises for Chapter 10: page 183

2. By Definition 110, $[\![\exists \upsilon \phi]\!]^{\alpha}_{\mathscr{A}} = T$ iff $[\![\phi]\!]^{\beta}_{\mathscr{A}} = T$ for at least one variable assignment β over $\mathscr{A}$ differing from α *at most* in its assignment to υ, iff $[\![\neg\phi]\!]^{\beta}_{\mathscr{A}} = F$ iff $[\![\neg\phi]\!]^{\beta}_{\mathscr{A}} \neq T$

iff $[\![\forall v \neg \phi]\!]^{\alpha}_{\mathscr{A}} \neq T$ (since β is a variable assignment differing from α at most in what it assigns to v), iff $[\![\neg \forall v \neg \phi]\!]^{\alpha}_{\mathscr{A}} = T$.

7. If $\Gamma, \phi[\tau/v] \vDash \psi$, then every $\mathscr{L}_2$-structure in which all of Γ and $\phi[\tau/v]$ are satisfied is also a structure in which ψ is satisfied.

   Suppose there is an $\mathscr{L}_2$-structure $\mathscr{A}$ in which all members of Γ are satisfied, and $\exists v \phi$ is satisfied, but ψ is not. Then there is a variable assignment α such that $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = T$. There exists an altered structure $\mathscr{A}'$ such that $[\![\tau]\!]^{\alpha}_{\mathscr{A}'} = \alpha(\tau)$ – i.e., it assigns to τ what α assigns to v in $\mathscr{A}$. Since $[\![\phi]\!]^{\alpha}_{\mathscr{A}} = T$, $[\![\phi[\tau/v]]\!]_{\mathscr{A}'}\, \alpha = T$. Suppose Γ is also satisfied in $\mathscr{A}'$ (if it is not, choose another altered structure in which it is: if there is none, then $\Gamma \cup \{\exists v \phi\}$ must be unsatisfiable, and the desired conclusion follows trivially). But then $\mathscr{A}'$ must make ψ true by hypothesis. On the other hand, since $\mathscr{A}'$ differs from $\mathscr{A}$ only in what it assigns to τ, and τ doesn't occur in ψ, ψ must also be false in $\mathscr{A}'$ since we've assumed it is false in $\mathscr{A}$. Contradiction: so $\Gamma, \exists v \phi \vDash \psi$.

10. Consider any $\mathscr{L}_2$-structure $\mathscr{A}$ which makes $\forall \xi(\phi \leftrightarrow \psi)$ true under some arbitrary variable assignment α. $\mathscr{A}$ is such that $[\![\phi \leftrightarrow \psi]\!]^{\beta}_{\mathscr{A}} = T$ for any variable assignments β differing from α at most in what they assign to ξ.

    Suppose $[\![\forall \xi \phi]\!]^{\alpha}_{\mathscr{A}} \neq [\![\forall \xi \psi]\!]^{\alpha}_{\mathscr{A}}$. Then for one such variable assignment β differing from α at most in what it assigns to ξ, $[\![\phi]\!]^{\beta}_{\mathscr{A}} \neq [\![\psi]\!]^{\beta}_{\mathscr{A}}$. But since $[\![\phi \leftrightarrow \psi]\!]^{\beta}_{\mathscr{A}} = T$, $[\![\phi]\!]^{\beta}_{\mathscr{A}} = [\![\psi]\!]^{\beta}_{\mathscr{A}}$. Contradiction; so $[\![\forall \xi \phi]\!]^{\alpha}_{\mathscr{A}} = [\![\forall \xi \psi]\!]^{\alpha}_{\mathscr{A}}$, and hence $[\![\forall \xi \phi \leftrightarrow \forall \xi \psi]\!]^{\alpha}_{\mathscr{A}} = T$.

15. An uncountable domain has too many members to be named; that was the problem for substitutional quantification. But since all of our formula are finite, no open formula of $\mathscr{L}_2$ ever has more than finitely many variables needed to be assigned values by any one variable assignment. So whenever we need to consider whether a sentence is true, we only need to consider variable assignments which temporarily assign names to finitely many members of the uncountable domain.

    But we have enough variable assignments so that any object which can be discussed even indirectly can be temporarily named. So consider the problem case $\forall x(\text{Integer}(x))$ evaluated in a structure $\mathscr{A}$ with domain of the reals $\mathbb{R}$ and an interpretation which assigns to every constant an integer. This quantified sentence is nevertheless false, since there is a variable assignment α on that structure which assigns to x some non-integer value, and $[\![\text{Integer}(x)]\!]^{\alpha}_{\mathscr{A}} = F$.

Exercises for Chapter 13: page 210

Appendix B

Greek letters

α, A	Alpha	ν, N	Nu
β, B	Beta	ξ, Ξ	Xi
γ, Γ	Gamma	o, O	Omicron
δ, Δ	Delta	π, Π	Pi
ϵ, E	Epsilon	ρ, P	Rho
ζ, Z	Zeta	σ, Σ	Sigma
η, H	Eta	τ, T	Tau
θ, Θ	Theta	υ, Υ	Upsilon
ι, I	Iota	ϕ, Φ	Phi
κ, K	Kappa	χ, X	Chi
λ, Λ	Lambda	ψ, Ψ	Psi
μ, M	Mu	ω, Ω	Omega

Table B.1: Table of Greek letters

List of Definitions

List of Tables

List of Figures

Bibliography

Beall, JC and van Fraassen, Bas C (2003), *Possibilities and Paradox*. Oxford: Oxford University Press.

—— and Restall, Greg, 'Logical Consequence'. In Edward N Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. URL plato.stanford.edu/entries/logical-consequence/.

Belnap, Jr., Nuel D (1977), 'A Useful Four-Valued Logic'. In J Michael Dunn and G Epstein (eds.), *Modern Uses of Multiple-Valued Logic*, Dordrecht: D Reidel, pp. 8–37.

Bennett, Jonathan (2003), *A Philosophical Guide to Conditionals*. Oxford: Oxford University Press.

Boolos, George S, Burgess, John P, and Jeffrey, Richard C (2007), *Computability and Logic*. Cambridge: Cambridge University Press, 5 ed.

Boolos, George (1971), 'The Iterative Conception of Set'. *Journal of Philosophy* **68**: 215–232.

Borkowski, L and Słupecki, J (1958), 'The Logical Works of J Łukasiewicz'. *Studia Logica*, vol. 8: pp. 7–56.

Bostock, David (1997), *Intermediate Logic*. Oxford: Oxford University Press.

Burgess, John P (2009), *Philosophical Logic*. Princeton: Princeton University Press.

Caplan, Ben (2006), 'Empty names'. In Robert Stainton and Alex Barber (eds.), *The Encyclopedia of Language and Linguistics: Philosophy and Language*, vol. 4, Oxford: Elsevier, pp. 132–136.

Craig, William (1957), 'Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory'. *Journal of Symbolic Logic*, vol. 22: pp. 269–85.

Davis, Wayne (2014), 'Implicature'. In Edward N Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. URL plato.stanford.edu/entries/implicature/.

Dummett, Michael (1983), 'The Philosophical Basis of Intuitionist Logic'. In Paul Benacerraf and Hilary Putnam (eds.), *Philosophy of Mathematics: Selected Readings*, Cambridge: Cambridge University Press, 2 ed., pp. 97–129.

Edgington, Dorothy (2014), 'Indicative Conditionals'. In Edward N Zalta (ed.),

The Stanford Encyclopedia of Philosophy. URL plato.stanford.edu/entries/conditionals/.

Fine, Kit (1975), 'Vagueness, Truth and Logic'. *Synthese* **30**: 265–300.

van Fraassen, Bas C. (1980), *The Scientific Image*. Oxford: Clarendon Press.

Gentzen, Gerhard (1969), 'Investigations into Logical Deduction'. In M E Szabo (ed.), *The Collected Papers of Gerhard Gentzen*, Amsterdam: North-Holland, pp. 68–131.

Gibbard, Allen (1981), 'Two Recent Theories of Conditionals'. In W L Harper, *et al.*, (eds.), *Ifs*, Dordrecht: D. Reidel, pp. 211–47.

Gowers, Timothy (ed.) (2008), *The Princeton Companion to Mathematics*. Princeton, NJ: Princeton University Press.

Grice, Paul (1989), 'Logic and Conversation'. In *Studies in the Way of Words*, Cambridge, MA: Harvard University Press.

Halbach, Volker (2010), *The Logic Manual*. Oxford: Oxford University Press.

Harris, John H (1982), 'What's so logical about the 'logical' axioms?' *Studia Logica*, vol. 41: pp. 159–71.

Henkin, Leon (1949), 'The Completeness of the First-Order Functional Calculus'. *Journal of Symbolic Logic*, vol. 14: pp. 159–66.

Henle, James M, Garfield, Jay L, and Tymoczko, Thomas (2011), *Sweet Reason*. Chichester: Wiley-Blackwell, 2nd ed.

Heyting, A (1956), *Intuitionism: An Introduction*. Amsterdam: North-Holland.

Hodges, Wilfrid (2001), *Logic*. London: Penguin, 2nd ed.

Iemhoff, Rosalie (2015), 'Intuitionism in the Philosophy of Mathematics'. In Edward N Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. plato.stanford.edu/entries/logic-intuitionistic/.

Jeffrey, Richard C (2006), *Formal Logic: Its Scope and Limits*. Indianapolis: Hackett, 4th ed, with a supplement by John P Burgess.

King, Jeff C (2003) 'Tense, Modality, and Semantic Values'. *Philosophical Perspectives*, vol. 17: pp. 195–246.

Kratzer, Angelika (2012) 'Conditionals'. In her *Modals and Conditionals*, Oxford: Oxford University Press, pp. 85–108.

Kripke, Saul (1976), 'Is There a Problem About Substitutional Quantification?' In Gareth Evans and John McDowell (eds.), *Truth and Meaning*, Oxford: Oxford University Press, pp. 324–419.

—— (1980), *Naming and Necessity*. Cambridge, MA: Harvard University Press.

—— (2013), *Reference and Existence*. Oxford: Oxford University Press.

Kunda, Ziva (1999), *Social Cognition: Making Sense of People*. Cambridge, MA: MIT Press.

Lambert, Karel (2001), 'Free Logics'. In Lou Goble (ed.), *The Blackwell Guide to Philosophical Logic*, Oxford: Blackwell.

Lewis, David (1973), *Counterfactuals*. Oxford: Blackwell.

—— (1984), 'Putnam's Paradox'. *Australasian Journal of Philosophy*, vol. 62: pp. 221–36.

Ludlow, Peter (2013), 'Descriptions'. In Edward N Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. URL plato.stanford.edu/entries/descriptions/.

Machover, Moshé (1996), *Set Theory, Logic, and Their Limitations*. Cambridge: Cambridge University Press.

McGee, Vann (1985), 'A Counterexample to *Modus Ponens*'. *The Journal of Philosophy*, vol. 82: pp. 462–71.

—— (2006), 'There's a Rule for Everything'. In Agustín Rayo and Gabriel Uzquiano (eds.), *Absolute Generality*, Oxford: Oxford University Press, pp. 179–202.

Partee, Barbara H (1973), 'Some Structural Analogies Between Tenses and Pronouns in English'. *Journal of Philosophy*, vol. 70: pp. 601–607.

——, ter Meulen, Alice and Wall, Robert E (1990), *Mathematical Methods in Linguistics*. Dordrecht: Kluwer.

Potter, Michael (2004), *Set Theory and Its Philosophy*. Oxford: Oxford University Press.

Prawitz, Dag (2006), *Natural Deduction: A Proof-theoretic study*. New York: Dover.

Priest, Graham (2006), *In Contradiction*. Oxford: Oxford University Press, 2nd ed.

—— (2008), *An Introduction to Non-Classical Logic*. Cambridge: Cambridge University Press, 2nd ed.

Prior, A N (1960), 'The Runabout Inference-Ticket'. *Analysis*, vol. 21: pp. 38–9.

Pullum, Geoffrey K (2008), 'Scooping the Loop Snooper: a proof that the Halting Problem is unsolvable'. URL www.ling.ed.ac.uk/~gpullum/loopsnoop.pdf.

Putnam, Hilary (1980), 'Models and Reality'. *Journal of Symbolic Logic*, vol. 45: pp. 464–82.

Quine, W V O (1940) *Mathematical Logic*. Boston, MA: Harvard University Press.

Restall, Greg (2000) *An Introduction to Substructural Logics*. London: Routledge.

Richard, Mark (1986) 'Quotation, grammar, and opacity', *Linguistics and Philosophy* 9: 383–403.

Russell, Bertrand (1956), 'On Denoting'. In *Logic and Knowledge*, London and New York: Routledge, pp. 41–56.

Shapiro, Stewart (2000) *Thinking About Mathematics*. Oxford: Oxford University Press.

—— (2013), 'Classical Logic'. In Edward N Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. URL plato.stanford.edu/entries/logic-classical/.

Sider, Theodore (2010), *Logic for Philosophy*. Oxford: Oxford University Press.

Smith, Nicholas J J (2012), *Logic: The Laws of Truth*. Princeton: Princeton University Press.

Smullyan, Raymond M (1968), *First-Order Logic*. New York: Springer-Verlag.

Stalnaker, Robert C (1975), 'Indicative Conditionals'. *Philosophia*, vol. : pp. 269–86.

Strawson, P F (1950), 'On Referring'. *Mind*, vol. 59: pp. 320–44.

Tarski, Alfred (1933), 'The Concept of Truth in Formalized Languages'. In *Logic, Semantics, Metamathematics*, Indianapolis: Hackett, 2nd ed., 1983, pp. 152–278.

—— (1936), 'On the Concept of Logical Consequence'. In *Logic, Semantics, Metamathematics*, Indianapolis: Hackett, 2nd ed., 1983, pp. 409–420.

Taylor, Barry (2006), *Models, Truth, and Realism*. Oxford: Oxford University Press.

Thomson, J F (1990), 'In Defense of ⊃'. *The Journal of Philosophy*, vol. 87: pp. 57–70.

Turing, A M (1937), 'On Computable Numbers, with an Application to the Entscheidungsproblem'. *Proceedings of the London Mathematical Society*, vol. 42: pp. 230–65.