

ELEMENTS OF DEDUCTIVE LOGIC

Elements of Deductive Logic

AN INTRODUCTION TO FORMAL LOGIC
AND ELEMENTARY METATHEORY

WITH EXERCISES

Antony Eagle

University of Adelaide



Elements of Deductive Logic by Antony Eagle (antonyeagle.org) is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.

To view a copy of this license, visit creativecommons.org/licenses/by-sa/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

This book is open source. \LaTeX 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 2017-03-23

Preface

This is a textbook covering the basics of formal logic and elementary meta-theory. 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).

This book is made available under a Creative Commons license (CC BY-SA 4.0), which allows users to share the work as they like, and to create derivative and modified works for any purpose, as long as the original creator is appropriately credited, and derivative works are licensed in the same way. I hope this will encourage any users of the book to adapt it to their local environments.

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

0	Introduction	1
1	Mathematical Induction and Proof	5
1.1	Inductive Proofs	5
1.2	Proof by <i>Reductio</i>	11
1.3	Intuitionism	12
	Further Reading	14
	Exercises	14
2	Set Theory	15
2.1	Sets	15
2.2	Relations	25
2.3	Functions	33
2.4	Size	38
	Further Reading	42
	Exercises	42
3	The Syntax of \mathcal{L}_1	46
3.1	Strings and Quotation	46
3.2	Sentences	48
3.3	Proofs About Syntax	51
3.4	Proof By Induction on Complexity	52

3.5	The Size of \mathcal{L}_1	54
	Exercises	55
4	The Semantics of \mathcal{L}_1	57
4.1	Semantics for \mathcal{L}_1	57
4.2	Truth Tables	62
4.3	Satisfaction, Entailment, and other Semantic Notions . . .	63
4.4	Meaning, Possibility, and Time	68
4.5	Entailment and the Connectives	70
4.6	Structural Rules	72
4.7	Substitution	73
4.8	Truth-functions	78
	Exercises	83
5	Metatheory of \mathcal{L}_1	85
5.1	Disjunctive Normal Form	85
5.2	Expressive Adequacy and Functional Completeness	89
5.3	Duality	94
5.4	Interpolation	97
5.5	Compactness	100
5.6	Alternative Proof of Compactness	105
5.7	Decidability	106
	Exercises	112
6	Natural Deduction in \mathcal{L}_1	116
6.1	Natural Deduction Proofs	116
6.2	Soundness	119
6.3	Completeness	121
6.4	A Little More Proof Theory	125
6.5	Axioms	129
	Further Reading	132
	Exercises	132

CONTENTS

7	The Syntax and Semantics of \mathcal{L}_2	136
7.1	Syntax of \mathcal{L}_2	136
7.2	Semantics of \mathcal{L}_2	139
7.3	Some Semantic Theorems About \mathcal{L}_2	142
7.4	Alternative Semantics for \mathcal{L}_2	147
	Further Reading	149
	Exercises	149
8	Natural Deduction in \mathcal{L}_2	151
8.1	Proofs in \mathcal{L}_2	151
8.2	Soundness of \mathcal{L}_2	154
8.3	Completeness of \mathcal{L}_2	155
8.4	Decidability and Undecidability	157
	Further Reading	163
	Exercises	163
9	Syntax, Semantics, and Proofs in $\mathcal{L}_=$	166
9.1	Identity	166
9.2	Numerical Quantification and the Theory of Definite Descriptions	169
9.3	Compactness and Cardinality	174
9.4	Further Directions of Research and Study in Logic	176
	Further Reading	177
	Exercises	178
10	Logic and Natural Language	181
10.1	Conditionals	181
10.2	Entailment	189
10.3	Designators	192
10.4	More on Relations	198
	Further Reading	200
	Exercises	200

ELEMENTS OF DEDUCTIVE LOGIC

A	Answers to Selected Exercises	203
B	Greek letters	216
C	List of Definitions	217
	Bibliography	219

List of Tables

4.1	Truth Table for the Standard Connectives	64
4.2	A truth-function table for \mathbf{c}	79
4.3	The four 1-place truth functions	80
5.1	De Morgan Equivalences	92
5.2	Sheffer Stroke and Peirce Arrow	94
5.3	Duality Illustrated Using Truth Tables	95
6.1	Natural Deduction Rules for $\mathcal{L}_{\neg, \rightarrow}$	118
6.2	Negation rules in \mathcal{L}_{DP}	126
8.1	New Natural Deduction Rules for \mathcal{L}_2	152
9.1	New Natural Deduction Rules for $\mathcal{L}_=$	168
B.1	Table of Greek letters	216

List of Figures

2.1	Reflexivity and Transitivity	27
2.2	Symmetry	28
2.3	Connectedness and Totality	30
2.4	Partial ordering of a set of sets by \subseteq	32
2.5	Properties of functions	37

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 \mathcal{L}_1 ,² a predicate language \mathcal{L}_2 , and a predicate language with identity $\mathcal{L}_=$. And the metalanguage is the mathematically enriched version of English we use to talk about formal languages.

The Structure of this Book I begin, in chapters 1 and 2, by outlining some mathematical tools – specifically, some elementary *set theory*, including the theory of functions – that will help us study and understand the formal languages which are our subject. In chapters 3 and 4, we'll 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

¹Such 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.

²This 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.

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. We'll take a break from metatheory in chapter 6, to introduce and talk about *formal derivations* in sentential logic, and introduce a particular kind of natural deduction derivation system. Then we'll prove two central results connecting derivability with validity: the *soundness* and *completeness* theorems for sentential logic. I also mention issues of computational implementation, in particular, the *decidability* of sentential logic. In chapter 7, we'll review *predicate* logic and (in chapter 8) its natural deduction derivation system, and we'll prove soundness for standard predicate logic. I sketch the proof of completeness, and briefly consider issues of decidability. In chapter 9, issues about predicate logic with identity are discussed, including the soundness of the natural deduction system we construct, and issues about numerical quantification and compactness are discussed. In the final chapter 10, we turn from metatheory to issues about 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 C.

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).

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

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

mathematical theory: necessary mathematical truths that are collectively sufficient, when supplemented by appropriate 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 8.) 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 $\mathbb{N} = 0, 1, 2, 3, \dots$, 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 ,

$$(1.1) \quad 0 + 1 + 2 + \dots + n = \frac{n(n+1)}{2}.$$

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: $0+1+\dots+n = (n(n+1))/2$.

$$\begin{aligned}
 (1.2) \quad (0+1+\dots+n) &= \frac{n(n+1)}{2} \\
 (1.3) \quad (0+1+\dots+n)+n+1 &= \left(\frac{n(n+1)}{2} \right) + n+1 \\
 (1.4) &= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} \\
 (1.5) &= \frac{(n+2)(n+1)}{2} \\
 (1.6) &= \frac{(n+1)((n+1)+1)}{2}
 \end{aligned}$$

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. \square

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 predecessors 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 (\dagger) , $F(n)$. By the definition of G , for all $m < n$, $F(m)$. So for all $m \leq n$, $F(n)$. 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 (\dagger) again, for all n , $F(n)$, as desired. \square

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. \square

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

²Consider, 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.)

- 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- $P)$. \square

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 ' $\text{not not } S$ ' 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 203.

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: $X = \{x_1, \dots, x_n, \dots\}$.

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 $n > 0$ occurrences 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 $Y \not\subseteq X$ (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 $\mathbf{X} = X_1, \dots, X_n, \dots$ is a set of sets, $\bigcap \mathbf{X} = X_1 \cap \dots \cap X_n \cap \dots$

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: $X \setminus Y = \{x \in X : x \notin Y\}$.

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

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 \emptyset . $Y \setminus Y$ is also empty; by extensionality, $X \setminus X = \emptyset = Y \setminus Y$; 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 $x \in \emptyset$ then $x \in X$ – for the empty set has no members at all. By the definition of subset, therefore, $\emptyset \subseteq X$, for any X .

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

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

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

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 , $\emptyset \in \wp(X)$; and $X \in \wp(X)$. The name ‘power set’ comes from this fact:

²Hint 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.

Theorem 6

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

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, \dots, 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 i th 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 $\wp(X)$. \square

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 $\{x, \{x, y\}\}$.

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* $\langle\langle x, y \rangle\rangle =_{\text{df}} \{\{x\}, \{x, y\}\}$.

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 $y = v$). \square

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 : $X \cap y = \emptyset$.

(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 $\{X\} \cap y = \emptyset$. Since the only member of $\{X\}$ is X , it must be that $\{X\} \cap X = \emptyset$. But if $X \in X$, then X and $\{X\}$ do have a member in common – namely, X – and their intersection isn't empty. So $X \notin X$. \square

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 $\langle\langle x_1, \dots, x_n \rangle\rangle$ denote an ordered n -tuple, for $n \geq 2$. 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.* $\langle\langle x_1, \dots, x_{n+1} \rangle\rangle =_{\text{df}} \langle \langle\langle x_1, \dots, x_n \rangle\rangle, x_{n+1} \rangle$.

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 $\langle\langle x, y, z \rangle\rangle = \langle \langle\langle x, y \rangle\rangle, z \rangle$. 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 §7.2):

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

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

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 $y \in D$. (\emptyset 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 $\underbrace{D \times \dots \times D}_n$.

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 9.

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 .

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

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

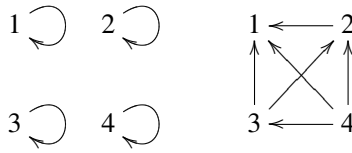


Figure 2.1: Reflexivity and Transitivity

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

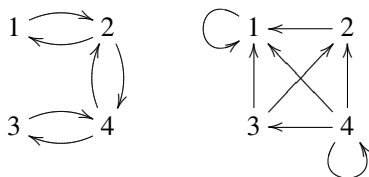


Figure 2.2: Symmetry

every object in the domain 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.

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

Now we may define:

Definition 26 (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 ,

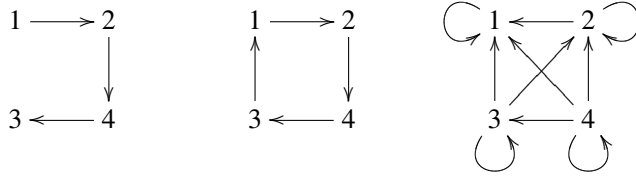


Figure 2.3: Connectedness and Totality

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 27 (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 28 (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 9

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 $\langle x, x \rangle \in R$ 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 . \square

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

Definition 29 (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 30 (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 10 (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

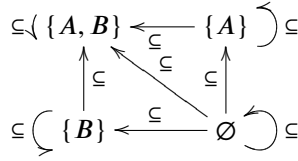
Definition 31 (Ordering). A relation R on D is a

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., \leq on \mathbb{N}). Also sometimes known as a *linear order*;

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


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

As defined above, a total order is reflexive, transitive, antisymmetric, and total. By Theorem 9, 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 32 (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: \leq is a well-ordering of the natural numbers \mathbb{N} . Here is another well-ordering of \mathbb{N} :

$$(2.1) \quad x \preceq y \quad \text{iff} \quad \begin{cases} x \text{ is even and } y \text{ is odd; or} \\ x \text{ and } y \text{ are even and } x \leq y; \text{ or} \\ x \text{ and } y \text{ are odd and } x \leq y. \end{cases}$$

The order induced by \preceq yields $0, 2, 4, \dots, 1, 3, 5, \dots$. In this ordering, every subset of \mathbb{N} has a \preceq -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 \leq 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 $\leq_{\mathbb{Z}}$. (This is strictly speaking a different relation, since $\langle -1, 0 \rangle \in \leq_{\mathbb{Z}}$ but $\langle -1, 0 \rangle \notin \leq$.) $\leq_{\mathbb{Z}}$ 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 $\leq_{\mathbb{Z}}$ (it does have a greatest member though, unlike \mathbb{N}). But \mathbb{Z} can be well-ordered: $x \leq_{\text{abs}} y$ iff either (i) $|x| \leq |y|$ or (ii) $|x| = |y|$ and $x \leq y$, where ‘ $|x|$ ’ here denotes the absolute value of x . This induces the order $0, -1, 1, -2, 2, \dots$

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 33 (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 10.

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 34 (Function). f is an n -place function iff (i) it is a relation (ii) and for any sequence x_1, \dots, x_n of objects, and any y, z , if $\langle x_1, \dots, x_n, y \rangle \in f$ and $\langle x_1, \dots, x_n, z \rangle \in f$, 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 35 (Domain). The *domain* (note: different to the domain of a relation) of a function f is the set

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

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

Definition 36 (Range). The *range* of f is the set

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

Definition 37 (Partial and Total). A function f is *partial* on domain D if $\mathcal{D}(f) \subset D$; it is *total* if $\mathcal{D}(f) = D$.

Obviously any function f is total on $\mathcal{D}(f)$, 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 38 (Operator). If f is a function, and $\mathcal{R}(f) \subseteq \mathcal{D}(f)$, then f is an *operator*.

Consider the numerical function $\text{sq} = \{\langle x, y \rangle : x^2 = y\}$, otherwise written $\text{sq}(x) = x^2$. 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, $\text{plus} = \{\langle x, y, z \rangle : z = x + y\}$. This is an arithmetical operator.

Some properties of functions are conveniently illustrated by plus.

Definition 39 (Commutative). A binary function f is commutative iff

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

Definition 40 (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 $\text{mean}(x, y) = \frac{x+y}{2}$. Clearly $\text{mean}(x, y) = \text{mean}(y, x)$. (This follows from the commutativity of plus.) But $\text{mean}(x, \text{mean}(y, z))$ is not in general equal to $\text{mean}(\text{mean}(x, y), z)$. 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 41 (Into, Onto, etc.). **Into** f is a function from X into Y iff $\mathcal{D}(f) = X$ and $\mathcal{R}(f) \subseteq Y$.

Onto f is a function from X onto Y iff $\mathcal{D}(f) = X$ and $\mathcal{R}(f) = Y$. 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.

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.)

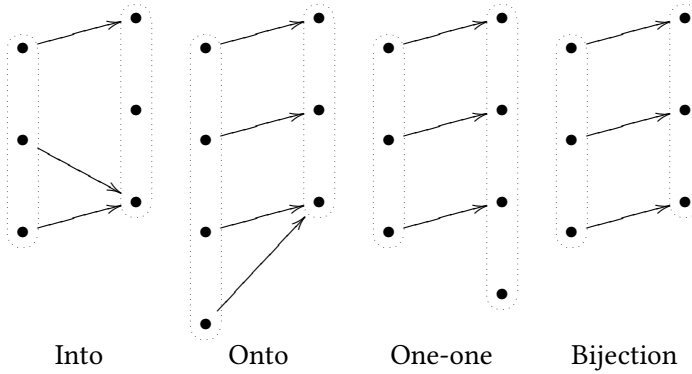


Figure 2.5: Properties of functions

- 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 complement. 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. 29 suffices to define an inverse function, but here is an equivalent definition directly in terms of functions:

Definition 42 (Inverse). If g is a function, it is the *inverse* of f iff $\mathcal{D}(f) = \mathcal{R}(g)$, $\mathcal{R}(f) = \mathcal{D}(g)$, 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 11

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

Proof. Left for exercises. □

Theorem 12

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. \square

2.4 Size

Definition 43 (Finite). A set X is finite iff there exists a set of natural numbers $N = \{1, \dots, n\}$ 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 $\{1, \dots, n\}$, 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 44 (Enumeration). f is an *enumeration* of X iff $\mathcal{D}(f) \subseteq \mathbb{N}$ (the natural numbers), $\mathcal{R}(f) = X$, 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 41. This is captured in the following definition:

Definition 45 ([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 46 (Equinumerosity). X and Y are *equinumerous* iff there exists a bijection from X to Y .

Definition 47 (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 47. But Definition 47 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 48 (\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 13

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

Proof. Obvious: by Definition 48, $\lambda \leq \mu$ iff there is an injection from X to Y iff there is a bijection from X to a subset of Y . \square

Theorem 14 (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 15 (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 48.

To show that $|X| < |\wp(X)|$, we need to show that $|X| \neq |\wp(X)|$, i.e. (by Definition 47), 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. \square

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 count-

able 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 16 (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. \square

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

Theorem 17 (Countable Unions)

If $\mathbf{X} = \{X_1, \dots, X_n, \dots\}$ 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 $\mathcal{R}(h_i) \cap \mathcal{R}(h_j) = \emptyset$.

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 $\mathcal{D}(h) = \bigcup \mathbf{X}$. Obviously $\mathcal{R}(h) \subseteq \mathbb{N}$. 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 philosophically 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) $X \not\subseteq X$.
- (d) If $X \subset Y$ then $Y \not\subseteq X$.
- (e) If $X \subset Y$ then $X \subseteq Y$.

2. Prove:

- (a) $X \cap X = X \cup X = X$.
- (b) $X \cap \emptyset = \emptyset$.
- (c) $X \cup \emptyset = X$.
- (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* $\langle x, y \rangle$ 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' 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.

SET THEORY

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 15 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 205.

Chapter 3

The Syntax of \mathcal{L}_1

3.1 Strings and Quotation

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

1. *Sentence letters* $P, Q, R, P_1, Q_1, R_1, P_2, Q_2, R_2, \dots$
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 \mathcal{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.

¹The 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 ' \sim ' where we use ' \neg ', ' \supset ' where we use ' \rightarrow ', and ' \equiv ' where we use ' \leftrightarrow '.

Definition 49 (String). A *string* in a language \mathcal{L} is any finite ordered sequence of characters from the alphabet of \mathcal{L} .

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 \mathcal{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:

‘ P_n ’, ‘ Q_n ’, ‘ R_n ’, for $n \geq 0$.

But this doesn’t work: for, according to our convention, ‘ 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 ‘ $n \geq 0$ ’. 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

²Of 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.

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.

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

Definition 50 (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 $n \geq 0$. 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 \mathcal{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 \mathcal{L}_1 , we may now specify our *syntax*. We do it by specifying which things are the (grammatical) sentences of our language.

Definition 51 (Sentences of \mathcal{L}_1). The *sentences* of \mathcal{L}_1 are those strings in the smallest collection including all strings satisfying these two conditions:

1. All sentence letters are sentences of \mathcal{L}_1 .
2. If ϕ and ψ are sentences of \mathcal{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 52 (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). \mathcal{L}_1 does not involve ternary or higher arity connectives.

Definition 53 (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 54 (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 \mathcal{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 \mathcal{L}_1 in chapter 4.

Abbreviatory Conventions When writing sentences of \mathcal{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 \mathcal{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 \mathcal{L}_1 . For an example of what we can show, consider this result:

Theorem 18 (Parenthesis Matching)

If ϕ is a sentence of \mathcal{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 51, every sentence is built up from the sentence letters using the subclauses of clause (2) of Definition 51. 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. \square

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 19 (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 18, 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 $\lceil \phi\psi \rceil$ 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 $\lceil \phi\psi \rceil$ 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 \mathcal{L}_1 . □

Informally, what Theorem 19 shows is that no sentence of \mathcal{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*.

Definition 55 (Complexity). The complexity of a sentence of \mathcal{L}_1 is the sum of the arities of the connectives occurring in it. (E.g., $\neg P$ has complexity 1, $\neg\neg(P \rightarrow Q)$ has complexity 4, etc.)

³The 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.

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 18: 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 20

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 \mathcal{L}_1

How big is the language \mathcal{L}_1 ? That is: what is the size of its set of sentences $S_{\mathcal{L}_1}$? There are infinitely many finite strings constructed from the alphabet of \mathcal{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 $\mathbf{P} \subset S_{\mathcal{L}_1}$, so $S_{\mathcal{L}_1}$ must be infinite too. The question is: is the size of $S_{\mathcal{L}_1}$ 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 \mathcal{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 \mathcal{L}_1 into corresponding sequences of natural numbers. And it turns out this entails the set of sentences of \mathcal{L}_1 is only countably infinite:

Lemma 21 (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:

⁴We need not do this, of course: we could use the alphabet consisting of P , Q , R , and the digits $0, \dots, 9$ 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 \mathcal{L}_1 .

if p_k is the k -th prime number, we map the sequence $\langle n_1, \dots, n_l \rangle$ to the product of the first l primes, each prime p_i raised to the power of n_i :

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

Since the coding function code is one-one, by the fundamental theorem of arithmetic (Gowers, 2008: §V.14), it has an inverse, which is from 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 17, the set of all finite sequences of natural numbers is countable. \square

By the Key Lemma, and the fact that we can model strings of \mathcal{L}_1 as finite sequences of natural numbers, there is a one-one correspondence between arbitrary finite strings of the alphabet of \mathcal{L}_1 and a countable set. Since the set of sentences of \mathcal{L}_1 $S_{\mathcal{L}_1}$ is a strict subset of the set of finite strings of the alphabet of \mathcal{L}_1 the cardinality of $S_{\mathcal{L}_1}$ is no greater than countable (by Theorem 13), 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 \mathcal{L}_1 has the same number of left and right parentheses.

3. Prove that if ϕ is a non-atomic \mathcal{L}_1 sentence, then there is exactly one formation clause from Definition 51 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 \mathcal{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 207.

Chapter 4

The Semantics of \mathcal{L}_1

4.1 Semantics for \mathcal{L}_1

To complete our specification of \mathcal{L}_1 , we need to provide some account of the intended meanings of sentences of \mathcal{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, \mathcal{L}_1 and other formal languages are not really devices for communication of meaning at all. They provide ways of representing struc-

¹We 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.

ture 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. \mathcal{L}_1 gives the connectives which combine with 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 \mathcal{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.

\mathcal{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 \mathcal{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 interpretation of \mathcal{L}_1 associates a *truth value* – either True or False – with every sentence letter of \mathcal{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. \mathcal{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 \mathcal{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 interpretation 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 \perp). 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. \mathcal{L}_1 is a classical logic, and any semantics for \mathcal{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 an assignment of truth values is known as a valuation:

Definition 56 (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 57 (\mathcal{L}_1 -Structure). An \mathcal{L}_1 -*structure* is a (total) function from the set of sentence letters of \mathcal{L}_1 into the set of classical truth values $\{T, F\}$.

An \mathcal{L}_1 -Structure is a formal mathematical rendering of what we informally termed an interpretation of \mathcal{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 \mathcal{L}_1 structure to a full classical valuation for \mathcal{L}_1 .

Definition 58 (\mathcal{L}_1 -valuation). When \mathcal{A} is any \mathcal{L}_1 -structure, $\llbracket \cdot \rrbracket_{\mathcal{A}}$ is a valuation function from the set of \mathcal{L}_1 sentences to the set of truth values $\{T, F\}$ iff it meets these conditions:

1. If ϕ is a sentence letter, $\llbracket \phi \rrbracket_{\mathcal{A}} = \mathcal{A}(\phi)$.
2. $\llbracket \neg \phi \rrbracket_{\mathcal{A}} = T$ if and only if (iff) $\llbracket \phi \rrbracket_{\mathcal{A}} = F$.
3. $\llbracket \phi \wedge \psi \rrbracket_{\mathcal{A}} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} = T$ and $\llbracket \psi \rrbracket_{\mathcal{A}} = T$.

4. $\llbracket \phi \vee \psi \rrbracket_{\mathcal{A}} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} = T$ or $\llbracket \psi \rrbracket_{\mathcal{A}} = T$ (or both).
5. $\llbracket \phi \rightarrow \psi \rrbracket_{\mathcal{A}} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} = F$ or $\llbracket \psi \rrbracket_{\mathcal{A}} = T$ (or both).
6. $\llbracket \phi \leftrightarrow \psi \rrbracket_{\mathcal{A}} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} = \llbracket \psi \rrbracket_{\mathcal{A}}$.

It is easy to see that for any \mathcal{L}_1 -structure \mathcal{A} the valuation function $\llbracket \cdot \rrbracket_{\mathcal{A}}$ 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 207 – that each sentence of \mathcal{L}_1 has exactly one main connective, so that every sentence of \mathcal{L}_1 falls under one and only one of the clauses in the definition of an \mathcal{L}_1 -valuation.)

What is also clear is that the definition of the valuation function specifies the meaning of the logical expressions of \mathcal{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 4.8.

Definition 59 (Agreement of Structures). When S is a set of \mathcal{L}_1 sentence letters, let us say that two \mathcal{L}_1 structures \mathcal{A} and \mathcal{B} *agree* on S iff for each ϕ in S , $\mathcal{A}(\phi) = \mathcal{B}(\phi)$.

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

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 22 (Falsity is Untruth)

$$\llbracket \phi \rrbracket_{\mathcal{A}} = F \text{ iff } \llbracket \phi \rrbracket_{\mathcal{A}} \neq T.$$

Proof. Base case: ϕ is a sentence letter. Because \mathcal{A} is a function, if $\llbracket \phi \rrbracket_{\mathcal{A}} = F$ then $\llbracket \phi \rrbracket_{\mathcal{A}} \neq T$. Because \mathcal{A} is into and total, if $\llbracket \phi \rrbracket_{\mathcal{A}} \neq T$ then $\llbracket \phi \rrbracket_{\mathcal{A}} = F$.

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 $\llbracket \phi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \neg\psi \rrbracket_{\mathcal{A}} = F$, iff $\llbracket \psi \rrbracket_{\mathcal{A}} = T$, iff $\llbracket \psi \rrbracket_{\mathcal{A}} \neq F$ iff $\llbracket \neg\psi \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} \neq T$.
2. Suppose $\phi = (\psi \vee \chi)$. Then $\llbracket \phi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \psi \vee \chi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \psi \rrbracket_{\mathcal{A}} = F$ and $\llbracket \chi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \psi \rrbracket_{\mathcal{A}} \neq T$ and $\llbracket \chi \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \psi \vee \chi \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} \neq T$.

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

4.2 Truth Tables

One way of representing \mathcal{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 \mathcal{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 58, 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 \mathcal{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: $P, Q, R, P_1, Q_1, R_1, \dots$. 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 59. (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 occurring 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 \mathcal{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.

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

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

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 \mathcal{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 60 (Satisfaction). Suppose Γ is any (possibly empty, possibly finite, possibly infinite) set of sentences of \mathcal{L}_1 , and \mathcal{A} is a \mathcal{L}_1 -structure, such that $\llbracket \gamma \rrbracket_{\mathcal{A}} = T$ for every sentence $\gamma \in \Gamma$. In that case, \mathcal{A} *satisfies* Γ , or \mathcal{A} is a *model* of Γ .

Definition 61 (Entailment). A set of sentences Γ (*semantically*) *entails* a sentence ϕ iff every \mathcal{L}_1 -structure which satisfies Γ also satisfies ϕ . Notation: $\Gamma \models \phi$.

Definition 62 (Tautology). ϕ is a *tautology* iff every \mathcal{L}_1 structure satisfies $\{\phi\}$. A tautology is sometimes called a *logical truth*.

Theorem 23

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

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

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

Theorem 24

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

Proof. ϕ is a tautology iff $\llbracket \phi \rrbracket_{\mathcal{A}} = T$ in every \mathcal{L}_1 structure \mathcal{A} . By Definition 58, this is the case iff $\llbracket \neg\phi \rrbracket_{\mathcal{A}} = F$ in every \mathcal{L}_1 structure \mathcal{A} ; iff no \mathcal{L}_1 structure satisfies $\{\neg\phi\}$, i.e., $\neg\phi$ is a contradiction. \square

Theorem 25 (Entailment and Unsatisfiability)

$\Gamma \models \phi$ iff $\Gamma, \neg\phi \models$.

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

This elementary theorem is nevertheless rather useful. One thing it shows is that *reductio* reasoning is good in \mathcal{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 $\Gamma, \phi \models$ 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 .

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

²We 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 ' $\Gamma \models$ ' is shorthand for ' Γ is unsatisfiable'.

Definition 63 (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.

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 \mathcal{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 \mathcal{L}_1 -structure is, in some ways, precisely the opposite of this. It is a way of interpreting the sentence letters of \mathcal{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).

Why does logic concern itself with what follows from a sentence under every possible reinterpretation of the non-logical expressions (involved (i.e., those without a fixed meaning - the sentence letters, in the case of \mathcal{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 \mathcal{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 \mathcal{L}_1 .

4.4 Meaning, Possibility, and Time

The picture we have of meaning in our language \mathcal{L}_1 is this. A sentence letter has as its meaning, relative to an \mathcal{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 \mathcal{L}_1 -*logically possible scenario* as one described by a satisfiable set of \mathcal{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 \mathcal{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 \mathcal{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 \mathcal{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.

³Even 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.

- 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 \mathcal{L}_1 sentences concern just what is *presently true*; if you wanted to handle claims about what *was true* (which we handle 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 \mathcal{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 \mathcal{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 P , letting t_1 correspond to some \mathcal{L}_1 structure in which P is assigned T , and let t_2 correspond to some \mathcal{L}_1 structure in which P is assigned F . A whole possible world, on this view, is then modelled by the sequence of structures – because a possible situation, to accommodate 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 \mathcal{L}_1 (and \mathcal{L}_2). As we’ll see when we move to talking about \mathcal{L}_2 later in this book, there are already natural language constructions which cannot be handled by \mathcal{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.

4.5 Entailment and the Connectives

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

Theorem 26 (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 $\{\gamma_1, \dots, \gamma_n\}$, then for every \mathcal{L}_1 structure \mathcal{A} , \mathcal{A} satisfies Γ iff $\llbracket (\gamma_1 \wedge \dots \wedge \gamma_n) \rrbracket_{\mathcal{A}} = T$.

Definition 64 (Equivalence). ϕ and ψ are *logically equivalent* iff $\phi \models \psi$ and $\psi \models \phi$.

Theorem 27 (Equivalence and Biconditional)

ϕ and ψ are logically equivalent iff $\models \phi \leftrightarrow \psi$.

Proof. $\phi \models \psi$ iff every structure in which ϕ has the value T is one in which ψ also has the value T , and *vice versa*. That is, iff in every structure \mathcal{A} , $\llbracket \phi \rrbracket_{\mathcal{A}} = \llbracket \psi \rrbracket_{\mathcal{A}}$. That holds in turn, by Definition 58, iff $\llbracket \phi \leftrightarrow \psi \rrbracket_{\mathcal{A}} = T$ in every structure, and $\phi \leftrightarrow \psi$ is a tautology. \square

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

Theorem 28 (Deduction)

$\Gamma, \phi \models \psi$ iff $\Gamma \models (\phi \rightarrow \psi)$.

Proof. $\Gamma, \phi \models \psi$ iff every \mathcal{L}_1 -structure which satisfies $\Gamma \cup \{\phi\}$ also satisfies $\{\psi\}$. That holds iff there is no structure \mathcal{A} in which Γ is satisfied and in which $\llbracket \phi \rrbracket_{\mathcal{A}} = T$ and $\llbracket \psi \rrbracket_{\mathcal{A}} = F$. By Definition 58, there is no structure \mathcal{A} in which Γ is satisfied while $\llbracket \phi \rightarrow \psi \rrbracket_{\mathcal{A}} = F$, i.e., $\Gamma \models \phi \rightarrow \psi$. \square

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

Corollary 29

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 $\Gamma \models \phi$ iff $\models (\gamma_1 \wedge \dots \wedge \gamma_n) \rightarrow \phi$.

Proof. Obvious from Theorem 26 and Theorem 28. \square

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 \mathcal{L}_1 something

analogous to entailment between sentences of \mathcal{L}_1 . The conditional ‘If we return our rental car late, we will be charged’ is true, but not tautologically 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 $(P \wedge Q \models P)$. We return to the topic of conditionals in chapter 10.

4.6 Structural Rules

The expression of entailment $\Gamma \models \phi$ 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 30 (Permutation)

$\Gamma, \psi, \chi, \Delta \models \phi$ iff $\Gamma, \chi, \psi, \Delta \models \phi$ (*premise order doesn't matter*).

Theorem 31 (Contraction)

$\Gamma, \psi, \psi, \Delta \models \phi$ iff $\Gamma, \psi, \Delta \models \phi$ (*duplicate premises don't matter*).

Theorem 32 (Weakening)

If $\Gamma \models \phi$, then $\Gamma, \psi \models \phi$.

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 . \square

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 33 (Cut)

If $\Gamma, \psi \models \phi$ and $\Gamma \models \psi$, then $\Gamma \models \phi$.

Proof. Assume that $\Gamma, \psi \models \phi$. 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: $\Gamma \models \phi$. \square

Theorem 34 (Transitivity)

If $\Gamma \models \psi$ and $\psi \models \phi$, then $\Gamma \models \phi$.

Theorem 35 (Contraposition)

$\phi \models \psi$ iff $\neg\psi \models \neg\phi$.

4.7 Substitution

Formality of Logic The notions of logical validity and entailment in \mathcal{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

⁴For 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 $\{a, a, b\}$ and $\{a, b\}$ 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 $\{P, P\}$ provides are strictly more than those provided by $\{P\}$. 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.

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 $P \rightarrow Q$ entail Q , yet R and $R \rightarrow P_1$ 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 \mathcal{L}_1 , the substitutable constituents are sentences (including sentence letters), and the logical structure is given by the logical connectives.

Definition 65 (Constituent sentence). A *constituent sentence* of ϕ is any well-formed \mathcal{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 66 (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 67 (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 36 (Substitution)

Suppose ϕ and θ are sentences, and s a sentence letter. For any structure \mathcal{A} , define a structure for every sentence letter α :

$$\mathcal{A}^\star(\alpha) = \begin{cases} \mathcal{A}(\alpha) & \text{iff } \alpha \neq s \\ \llbracket \theta \rrbracket_{\mathcal{A}} & \text{iff } \alpha = s. \end{cases}$$

Then $\llbracket \phi \rrbracket_{\mathcal{A}^\star} = \llbracket \phi[\theta/s] \rrbracket_{\mathcal{A}}$.

Proof. The new structure \mathcal{A}^\star is just like \mathcal{A} , with the possible exception that it assigns to s the value that θ has in \mathcal{A} .

Base case: If ϕ is a sentence letter, then the result follows by construction of \mathcal{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 $\llbracket \phi_i \rrbracket_{\mathcal{A}^\star} = \llbracket \phi_i[\theta/s] \rrbracket_{\mathcal{A}}$ for each ϕ_i which is a constituent of ϕ .

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

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

The desired result follows: $\llbracket \phi \rrbracket_{\mathcal{A}^\star} = \llbracket \phi[\theta/s] \rrbracket_{\mathcal{A}}$. *Mutatis mutandis* for the other connectives. \square

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

Theorem 37 (Substitution of Material Equivalents)

If $\llbracket \theta \rrbracket_{\mathcal{A}} = \mathcal{A}(s)$, then $\llbracket \phi \rrbracket_{\mathcal{A}} = \llbracket \phi[\theta/s] \rrbracket_{\mathcal{A}}$.

Proof. If $\llbracket \theta \rrbracket_{\mathcal{A}} = \mathcal{A}(s)$, then $\mathcal{A} = \mathcal{A}^*$; applying Theorem 36, the result follows immediately. \square

Theorem 38 (Substitution of Sentence Letters)

If Γ entails ϕ , then so too $\Gamma[\theta/s] \models \phi[\theta/s]$, for any θ, s .

Proof. Now suppose for *reductio* that $\Gamma \models \phi$, but $\Gamma[\theta/s] \not\models \phi[\theta/s]$. In that case, there is a structure \mathcal{B} such that for each $\gamma \in \Gamma$, $\llbracket \gamma[\theta/s] \rrbracket_{\mathcal{B}} = T$, but $\llbracket \phi[\theta/s] \rrbracket_{\mathcal{B}} = F$.

By the Substitution theorem 36, there is therefore a structure \mathcal{B}^* such that for each $\gamma \in \Gamma$, $\llbracket \gamma \rrbracket_{\mathcal{B}^*} = T$ and $\llbracket \phi \rrbracket_{\mathcal{B}^*} = F$. But in that case, $\Gamma \not\models \phi$ after all, since there is an \mathcal{L}_1 structure where the premises are true and the conclusion false, contradicting our *reductio* supposition, which must have been wrong. \square

Generalised Substitution and Formal Logic A generalisation of Theorem 38 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 39 (General Substitution)

When Γ is finite, if $\Gamma \models \phi$, then also $\Gamma[\theta/\chi] \models \phi[\theta/\chi]$, for any θ, χ .

Proof. Suppose for *reductio* that $\Gamma \models \phi$, but there is a sentence letter s not occurring in Γ or ϕ such that $\Gamma[s/\chi] \not\models \phi[s/\chi]$. There must be a structure \mathcal{B}^* such that for each $\gamma \in \Gamma$, $\llbracket \gamma[s/\chi] \rrbracket_{\mathcal{B}^*} = T$, but $\llbracket \phi[s/\chi] \rrbracket_{\mathcal{B}^*} = F$. By theorem 36, there is therefore a structure \mathcal{B} such that for each $\gamma \in \Gamma$, $\llbracket \gamma \rrbracket_{\mathcal{B}} = T$ and $\llbracket \phi \rrbracket_{\mathcal{B}} = F$.⁵ But in that case, $\Gamma \not\models \phi$ after all, contradiction: so our

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

supposition must have been wrong. So in fact for any sentence letter s not occurring in Γ or ϕ , when $\Gamma \models \phi$, $\Gamma[s/\chi] \models \phi[s/\chi]$. 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 38 to show that $\Gamma[s/\chi][\theta/s] \models \phi[s/\chi][\theta/s]$. But this latter sequent just is $\Gamma[\theta/\chi] \models \phi[\theta/\chi]$. \square

Theorem 39 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 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 $P, P \rightarrow Q \models Q$, accordingly so too $P, (P \rightarrow \neg P) \models \neg P$, 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-substitution 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.

⁶The 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 55), which we will prove in chapter 5, assures us that: $\Gamma \models \phi$ iff there is some finite set $\Gamma^- \subseteq \Gamma$ such that $\Gamma^- \models \phi$. We can then apply Theorem 39 to Γ^- . But don’t worry about this wrinkle until then; we rarely come across arguments with infinitely many premises in everyday life....

⁷So $(P \vee Q)$ 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 $(P \vee Q)[Q/P] = (Q \vee Q)$, $(Q \vee Q)$ does not share a logical form with $(P \vee Q)$, since we cannot uniformly substitute for Q in $(Q \vee Q)$ and obtain $(P \vee Q)$.

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 40 (Equivalents)

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

Proof. Left for exercise. □

Theorem 41 (Equivalence)

Suppose ϕ is a constituent sentence of χ , and $\chi' = \chi[\psi/\phi]$. Then $\phi \leftrightarrow \psi \models \chi \leftrightarrow \chi'$.

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

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

4.8 Truth-functions

Definition 68 (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 \mathcal{L}_1 -structures. One such truth-function table is depicted in Table 4.2, for the

		c
<i>T</i>	<i>T</i>	<i>T</i>
<i>T</i>	<i>F</i>	<i>F</i>
<i>F</i>	<i>T</i>	<i>T</i>
<i>F</i>	<i>F</i>	<i>T</i>

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

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 42

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 *F*s and *T*s 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. \square

Consider the 1-place truth functions. By Theorem 42, 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 \mathbf{c} in Table 4.2 bears some resemblance to the truth table for $(P \rightarrow Q)$, 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 69 (Expression). A sentence ϕ expresses a truth function f_n iff if ϕ contains exactly the sentence letters s_1, \dots, s_n (ordered in the standard way) then for every \mathcal{L}_1 -structure \mathcal{A} ,

$$f(\mathcal{A}(s_1), \dots, \mathcal{A}(s_n)) = \llbracket \phi \rrbracket_{\mathcal{A}}.$$

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

Since every \mathcal{L}_1 sentence has a truth-table produced in the orthodox way, every \mathcal{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 89).

	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 \mathcal{L}_1 (Table 4.1).

Definition 70 (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 ϕ_1, \dots, ϕ_n ,

$$\llbracket \oplus(\phi_1, \dots, \phi_n) \rrbracket_{\mathcal{A}} = f(\llbracket \phi_1 \rrbracket_{\mathcal{A}}, \dots, \llbracket \phi_n \rrbracket_{\mathcal{A}}),$$

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

It is evident from Table 4.1 that each of the connectives of \mathcal{L}_1 are truth-functional connectives, so \mathcal{L}_1 is a *truth-functional language*. We've already encountered the 1-place truth function corresponding to the negation connective: it is **n** from Table 4.3. We also encountered the 2-place connective **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 $\lceil \text{Necessarily } \phi \rceil$. If ϕ is false in a structure, that obviously guarantees that $\lceil \text{Necessarily } \phi \rceil$ 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 $\lceil \text{Necessarily } \phi \rceil$ 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 \mathcal{L}_1 structures and classical valuations. That is the topic taken up by *modal logic* – see the discussion at the end of chapter 9. We already can see, however, one expressive limitation of \mathcal{L}_1 as compared to English: for English includes non-truth-functional connectives like ‘necessarily’, ‘probably’, ‘typically’, etc., and \mathcal{L}_1 does not. Every sentence in \mathcal{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 71 (Alternative \mathcal{L}_1 -valuation). When \mathcal{A} is any \mathcal{L}_1 -structure, the valuation function generated by \mathcal{A} is this function:

$$\llbracket \phi \rrbracket_{\mathcal{A}} = \begin{cases} \mathcal{A}(\phi) & \text{if } \phi \text{ is a sentence letter;} \\ \mathbf{n}(\llbracket \psi \rrbracket_{\mathcal{A}}) & \text{if } \phi = \neg\psi; \\ \mathbf{k}(\llbracket \psi \rrbracket_{\mathcal{A}}, \llbracket \chi \rrbracket_{\mathcal{A}}) & \text{if } \phi = (\psi \wedge \chi); \\ \mathbf{a}(\llbracket \psi \rrbracket_{\mathcal{A}}, \llbracket \chi \rrbracket_{\mathcal{A}}) & \text{if } \phi = (\psi \vee \chi); \\ \mathbf{c}(\llbracket \psi \rrbracket_{\mathcal{A}}, \llbracket \chi \rrbracket_{\mathcal{A}}) & \text{if } \phi = (\psi \rightarrow \chi); \\ \mathbf{e}(\llbracket \psi \rrbracket_{\mathcal{A}}, \llbracket \chi \rrbracket_{\mathcal{A}}) & \text{if } \phi = (\psi \leftrightarrow \chi). \end{cases}$$

This definition is obviously equivalent to that in Definition 58, in that both definitions determine the same valuation of all the sentences of the language given the same initial structure.

Definition 72 (Base). (Beall and van Fraassen, 2003: 27) Let $\llbracket \cdot \rrbracket_{\mathcal{A}}$ be a classical valuation. A *base* \mathfrak{B} for $\llbracket \cdot \rrbracket_{\mathcal{A}}$ 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 38), and c is a function from the set of connectives of \mathcal{L}_1 , $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, into \mathbb{O} , such that

- $\llbracket \phi \rrbracket_{\mathcal{A}} \in V$, for any \mathcal{L}_1 sentence ϕ ;
- for any sentences ϕ_1, \dots, ϕ_n , and any n -ary connective \oplus ,

$$\llbracket \oplus(\phi_1, \dots, \phi_n) \rrbracket_{\mathcal{A}, \mathcal{L}} = c(\oplus)(\llbracket \phi_1 \rrbracket_{\mathcal{A}, \mathcal{L}}, \dots, \llbracket \phi_n \rrbracket_{\mathcal{A}, \mathcal{L}}).$$

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 70). A base for \mathcal{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 $\Gamma \models$ iff for every $\gamma \in \Gamma$, $\Gamma \setminus \{\gamma\} \models \neg\gamma$. (Hint: Theorem 25.)
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) Prove Transitivity.
 - (b) Prove Contraposition.
 - (c) Prove this rule, related to Cut: if $\Gamma \models \phi$, and $\phi, \Delta \models \psi$, then $\Gamma, \Delta \models \psi$.
4. Show the remaining clauses of the induction step for the proof of Theorem 22 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 40).
6. (a) Prove the remaining induction cases in the proof of Equivalence (Theorem 41).
 (b) As a corollary of the Equivalence Theorem, prove that if $\models \phi \leftrightarrow \psi$ and ϕ is a constituent sentence of χ , then $\models \chi \leftrightarrow \chi[\psi/\phi]$.
7. Show that, if $\phi, \psi \models \chi$ is a false sequent, it has a substitution instance $\phi', \psi' \models \chi'$, 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 39] that successive chains of uniform substitutions never allow one to transform a correct sequent to an incorrect one.)
8. Show that if Γ contains n sentence letters, its truth table has 2^n rows.
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 \mathcal{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):

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

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

Answers to selected exercises on page 208.

Chapter 5

Metatheory of \mathcal{L}_1

5.1 Disjunctive Normal Form

Definition 73 (Arbitrary Conjunction). An inductive definition:

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

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

Definition 75 (Disjunctive Normal Form (DNF)). An \mathcal{L}_1 sentence of ϕ is in *disjunctive normal form* iff there exist n, m_1, \dots, 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 DNF also. Another way of characterising which \mathcal{L}_1 sentences are in DNF is as follows: a sentence is DNF iff 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 43 (DNF)

Every truth function is expressed by a \mathcal{L}_1 sentence in DNF.

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

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

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

For every structure \mathcal{A} , define $c_{\mathcal{A}} = S_1^{\mathcal{A}} \wedge \dots \wedge S_n^{\mathcal{A}} = \left[\bigwedge_{i=1}^n S_i^{\mathcal{A}} \right]$. Again, it is obvious that $\llbracket c_{\mathcal{A}} \rrbracket_{\mathcal{A}} = T$, since it is a conjunction of true conjuncts. It is equally obvious that for any structure \mathcal{B} that doesn't agree with \mathcal{A} on the sentence letters s_1, \dots, s_n , $\llbracket c_{\mathcal{A}} \rrbracket_{\mathcal{B}} = F$, 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, \dots, s_n , there are only finitely many distinct sentences $c_{\mathcal{A}}$ for various \mathcal{A} . Therefore, there are only finitely many $c_{\mathcal{A}}$ such that \mathcal{A} is an f -agreeing structure; let them be enumerated c_1, \dots, c_m . It is obvious that each c_i is true in one and only one structure, and that each such

structure is f -agreeing.

Define $\mathfrak{d} = c_1 \vee \dots \vee c_m = \left[\bigvee_{j=1}^m c_j \right]$, *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 c_i is true, i.e., \mathfrak{d} is true in any f -agreeing structure.

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

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 \mathcal{A} is $\llbracket P \rrbracket_{\mathcal{A}} = T$, $\llbracket Q \rrbracket_{\mathcal{A}} = F$, 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 44

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. \square

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 $f(T, \dots, T) = T$ (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 45

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

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 \rightarrow and \wedge . \square

5.2 Expressive Adequacy and Functional Completeness

Expressive Adequacy and Functional Completeness

Definition 76 (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 \mathcal{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 \mathcal{L}_1 sentences in DNF form are collectively expressively adequate, \mathcal{L}_1 is expressively adequate.

Definition 77 (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

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

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 \mathcal{L}_1 sentence, and so \mathcal{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 78 (Maximum and Minimum). The *maximum* of a sequence of truth values is defined inductively:

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

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

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

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

Theorem 46 (Functional Completeness of $\{\mathbf{n}, \mathbf{k}, \mathbf{a}\}$)

If $f(x_1, \dots, x_n)$ 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 occurrence of \mathbf{n} in the function definition.

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 *dispensable* – 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 \mathcal{L}_1 sentence expresses a truth-function, suffices to show that every \mathcal{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 dispensable – 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 ' $\phi \models \psi$ ' abbreviate ' $\phi \models \psi$ and $\psi \models \phi$ ' – that is, that ϕ and ψ are logically equivalent.

Theorem 47 (De Morgan Equivalences)

$$(\phi \vee \psi) \models \neg(\neg\phi \wedge \neg\psi),$$

$$(\phi \wedge \psi) \models \neg(\neg\phi \vee \neg\psi).$$

²It 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). \square

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 48 (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 47 without use of a truth table.³ This result also shows that $\{\mathbf{n}, \mathbf{k}\}$ and $\{\mathbf{n}, \mathbf{a}\}$ are functionally complete.

³Consider the equivalence of $(\phi \vee \psi)$ and $\neg(\neg\phi \wedge \neg\psi)$. By Theorem 48, $\mathbf{a}(x, y) = \mathbf{n}(\mathbf{k}(\mathbf{n}(x), \mathbf{n}(y)))$, so for any \mathcal{A} ,

$$\begin{aligned} \llbracket \phi \vee \psi \rrbracket_{\mathcal{A}} &= \mathbf{a}(\llbracket \phi \rrbracket_{\mathcal{A}}, \llbracket \psi \rrbracket_{\mathcal{A}}) \\ &= \mathbf{n}(\mathbf{k}(\mathbf{n}(\llbracket \phi \rrbracket_{\mathcal{A}}), \mathbf{n}(\llbracket \psi \rrbracket_{\mathcal{A}}))) \\ &= \mathbf{n}(\mathbf{k}(\llbracket \neg\phi \rrbracket_{\mathcal{A}}, \llbracket \neg\psi \rrbracket_{\mathcal{A}})) \\ &= \mathbf{n}(\llbracket (\neg\phi \wedge \neg\psi) \rrbracket_{\mathcal{A}}) \\ &= \llbracket \neg(\neg\phi \wedge \neg\psi) \rrbracket_{\mathcal{A}}. \end{aligned}$$

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 apart from whether it can express a 2- or more place truth function, consider the 1-place function t which is constantly true. Since $t(T) = t(F)$, we would need a sentence ϕ involving only negation such that in even one case $\llbracket \phi \rrbracket_{\mathcal{A}} = \llbracket \neg\phi \rrbracket_{\mathcal{A}}$; 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 $\{n\}$ is functionally incomplete; since \neg cannot express t , so t can't be constructed solely from n .

Given our definition of a connective, we can imagine extending our language \mathcal{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 T – this is like a sentence letter with a fixed interpretation, T in every structure. As we know from the DNF theorem, adding T doesn't add to the expressive power of the language: T is logically equivalent to $(P \vee \neg P)$, and so adding it would be redundant. Adding \wedge to the language \mathcal{L}_{\neg} , in which the only connective is negation, is not redundant: $\mathcal{L}_{\neg, \wedge}$ is functionally complete and \mathcal{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.

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

Definition 79 (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 80 (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 5.3⁴).

Other connectives have duals too: $\phi \leftrightarrow \psi$ is dual to the operator \leftrightarrow (where $\phi \leftrightarrow \psi$ is true just in case ϕ and ψ have different truth values). Negation \neg is an interesting 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 \mathcal{L}_1 which involves only connectives in $\{\wedge, \vee, \neg\}$ (any sentence of \mathcal{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 substituting $\neg s$ for each sentence letter s in ϕ (even those already negated).
2. Let ϕ^* be the sentence that results from replacing each connective in ϕ by its dual.

Note to begin that $\neg \neg \overline{\phi} = \overline{\phi}$ and $\overline{\phi \wedge \psi} = \overline{\phi} \vee \overline{\psi}$, 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.

ϕ	ψ	$\phi \wedge \psi$	$\phi \uparrow \psi$		ϕ	ψ	$\phi \vee \psi$	$\phi \downarrow \psi$
T	T	T	F	\Leftrightarrow	F	F	F	T
T	F	F	F		F	T	T	T
F	T	F	F		T	F	T	T
F	F	F	T		T	T	T	F

Table 5.3: Duality Illustrated Using Truth Tables

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

Lemma 49 (Duality for Conjunction and Disjunction)

For any ϕ , $\phi^* \models \neg\bar{\phi}$.

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

Induction step:

1. ϕ is $\neg\psi$, and the hypothesis holds for ψ . Then $\phi^* = (\neg\psi)^* = \neg(\psi^*)$, because \neg is its own dual. Because the theorem holds of ψ , $\neg(\psi^*) \models \neg\neg\bar{\psi}$; but $\neg\neg\bar{\psi} = \neg\bar{\bar{\psi}} = \neg\bar{\phi}$.
2. ϕ is $(\psi \wedge \chi)$, and the hypothesis holds for ψ, χ . $\phi^* = (\psi \wedge \chi)^* = \psi^* \vee \chi^*$. By the induction hypothesis, $\psi^* \vee \chi^* \models \neg\bar{\psi} \vee \neg\bar{\chi}$. By De Morgan, $\neg\bar{\psi} \vee \neg\bar{\chi} \models \neg(\bar{\psi} \wedge \bar{\chi})$, which is $\neg(\bar{\psi \wedge \chi}) = \neg(\bar{\phi})$.
3. ϕ is $(\psi \vee \chi)$; much as for \wedge . □

Theorem 50 (Duality for Conjunction and Disjunction)

If ϕ^* is dual to ϕ and ψ^* is dual to ψ , then $\phi \models \psi$ iff $\psi^* \models \phi^*$.

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

Corollary 51

If $\phi \models \psi$, $\phi^* \models \psi^*$.

Each De Morgan equivalence in Theorem 47 follows from the other, given Corollary 51.

Theorem 52 (Duality for Tautologies)

If $\models \phi$, then $\models \neg(\phi^*)$.

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

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 $\mathbb{C} = \{\oplus_1, \dots, \oplus_n\}$ is *self-dual* iff $\mathbb{C} = \{\oplus_1^\star, \dots, \oplus_n^\star\}$. 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 \mathcal{A} is a structure, let \mathcal{A}^\star be the structure such that for all ϕ , $\mathcal{A}^\star(\phi) = T^\star$ iff $\mathcal{A}(\phi) = T$. Consider just those languages \mathcal{L} such that a connective \oplus is in \mathcal{L} iff \oplus^\star is in \mathcal{L} . (\mathcal{L}_1 is among these languages.) The dual ϕ^\star of a sentence ϕ of \mathcal{L} results from replacing *every* connective in ϕ by its dual. Then we can show:

$$\llbracket \phi \rrbracket_{\mathcal{A}} = \mathbf{n} \left(\llbracket \phi^\star \rrbracket_{\mathcal{A}^\star} \right).$$

5.4 Interpolation

Craig Interpolation Theorem for Sentential Logic

Theorem 53 (Craig Interpolation)

If $\phi \models \psi$, 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 $\phi \models \chi$ and $\chi \models \psi$, and each sentence letter in χ is in B .

Proof. Let the members of B be enumerated b_1, \dots, b_n .

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

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

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 \mathcal{B}_i on B).
- Moreover, χ is entailed by ϕ . Suppose ϕ is true in some structure \mathcal{C} , but χ is not. By definition, χ is false in \mathcal{C} because (i) either ψ is false in \mathcal{C} , or (ii) ψ is false in a structure just like \mathcal{C} in what it assigns to the members of B . Option (i) is excluded, because $\phi \models \psi$. So if χ is false in \mathcal{C} , it must be because some structure agreeing with \mathcal{C} on B makes ψ false. This structure need differ from \mathcal{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 \mathcal{C} and false in the other structure. But then the other structure need not differ from \mathcal{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 $\phi \models \psi$, ψ is both true and false in that structure. Contradiction; so there is no such \mathcal{C} , by *reductio*. But \mathcal{C} was arbitrary, so $\phi \models \chi$. \square

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

Corollary 54

If ϕ is logically equivalent to ψ , and there is a non-empty set B of sentence letters occurring 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 B .

Proof. Straightforward from transitivity of entailment and Theorem 53. \square

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 hygienic 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 observational 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 53 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 81 (Compactness). An interpreted language \mathcal{L} is said to be *compact* iff any unsatisfiable infinite set of sentences of \mathcal{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

⁵Each 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 .

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 \mathcal{L}_1 Is \mathcal{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 \mathcal{L}_1 , that \mathcal{L}_1 is in fact compact. In the next chapter, after we consider a derivation system for \mathcal{L}_1 , we can use the adequacy of that system to provide another more indirect proof of compactness for \mathcal{L}_1 (p. 125).

Compactness for \mathcal{L}_1 has the immediate consequence that if $\Gamma \models \phi$, then there is some finite set of premises Γ' such that $\Gamma' \models \phi$; every valid argument in \mathcal{L}_1 with 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 \mathcal{L}_1 that is validly entailed by infinitely many premises formulated in \mathcal{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. 106).

Definition 82 (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 55 (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_i s 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, \dots

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_i s are finitely satisfiable

Lemma 56

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 , $i \leq n$, 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 $\Delta, s_{n+1} \models$ and $\Theta, \neg s_{n+1} \models$. By weakening, both $\Delta, \Theta, s_{n+1} \models$ and $\Delta, \Theta, \neg s_{n+1} \models$.

But $\Delta \cup \Theta$ is a finite subset of a finitely satisfiable set, so is consistent. In any structure \mathcal{A} in which all the members of $\Delta \cup \Theta$ are true, either

$\llbracket s_{n+1} \rrbracket_{\mathcal{A}} = T$ or $\llbracket \neg s_{n+1} \rrbracket_{\mathcal{A}} = T$. So either $\Delta, \Theta, s_{n+1} \not\models$ or $\Delta, \Theta, \neg s_{n+1} \not\models$. Contradiction; so Γ_{n+1} is finitely satisfiable. \square

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

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

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

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

is a subset of Γ_k and is consistent.

By construction, for any k , and for all $\sigma \in S_k$, $\llbracket \sigma \rrbracket_{\mathcal{G}} = T$. Moreover, every structure \mathcal{G}' which agrees with \mathcal{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: \mathcal{G} satisfies Γ Suppose \mathcal{G} does not satisfy Γ . Then for some $\gamma \in \Gamma$, $\llbracket \gamma \rrbracket_{\mathcal{G}} = F$.

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 \mathcal{G} on every s_i , $i \leq k$, 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 \mathcal{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, \dots is an enumeration of the sentence letters occurring as constituents of sentences in Γ . Let S_n be the set consisting of the first n sentence letters in Γ under the enumeration, i.e., $S_n = \{s_1, s_2, \dots, s_n\}$. (S_0 is the empty set.)

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

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

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

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

Obviously \mathcal{A}_0 is 0-acceptable; to be 0-acceptable is for each finite subset of Γ to agree with \mathcal{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 \mathcal{A}_0 on that set.

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

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

Suppose \mathcal{A} isn't a model for Γ . Then there is a $\phi \in \Gamma$ such that $\llbracket \phi \rrbracket_{\mathcal{A}} = F$. Let s_k be the highest numbered sentence letter occurring in ϕ . Then \mathcal{A}_k must make ϕ true, since $\{\phi\}$ is a finite subset of Γ and \mathcal{A}_k is k -acceptable. But then every structure which agrees with \mathcal{A}_k on \mathbf{S}_k will make ϕ true; and \mathcal{A} is one such agreeing structure. Contradiction – \mathcal{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. 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 57 (Decidability of Finite Validity)

If Γ is a finite set of sentences of \mathcal{L}_1 , then it is decidable whether $\Gamma \models$.

Proof. Suppose Γ is a finite set of \mathcal{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 $\Gamma \models$?’. 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 $\Gamma \models \phi$ (just check whether $\Gamma, \neg\phi \models$ by Theorem 25), and thus that *validity is decidable*.

Definition 83 (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 $n \notin N$ in a finite time.

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 \mathcal{L}_1 sentences and numbers established by the code in Lemma Theorem 21, we can by courtesy extend the notion of a recursive set of sets of \mathcal{L}_1 sentences in the obvious way: a set of \mathcal{L}_1 sentences is recursive iff there is an effectively procedure which decides whether an arbitrary \mathcal{L}_1 sentence is a member of it or not. Theorem 57 shows that the set of \mathcal{L}_1 contradictions is recursive. Any \mathcal{L}_1 sentence is equivalent to a conjunction, by the CNF theorem. An contradiction is thus equivalent to some contradictory conjunction $\gamma_1 \wedge \dots \wedge \gamma_n$, where each γ_i is a disjunction of literals. But a conjunction $\gamma_1 \wedge \dots \wedge \gamma_n$ is a contradiction iff the finite set $\{\gamma_1, \dots, \gamma_n\}$ 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 \mathcal{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 84 (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 44 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 $n \notin N$, 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 $n = p_{i+1} - p_i$. 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 isn't among the prime gaps.

We can extend the notion of recursive enumerability to sentences of \mathcal{L}_1 too, and finite sets of sentences of \mathcal{L}_1 . Lemma 21 gives us a representation of \mathcal{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 \mathcal{L}_1 sentences by a number; and we may then say that a set of \mathcal{L}_1 sentences, or a set of sets of \mathcal{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 \mathcal{L}_1 sentences. For the number of effective procedures is countable; but the number of infinite sets of \mathcal{L}_1 sentences is uncountable (being a powerset of the countable set of all \mathcal{L}_1 sentences). So not every set Γ can be effectively listed. But those whose members can be effectively listed – the *recursively enumerable* sets – can also be effectively checked for unsatisfiability.

Theorem 58 (Positive Decidability of Unsatisfiability)

If Γ is a recursively enumerable set of \mathcal{L}_1 sentences, then if $\Gamma \models$, there is an effective procedure that will demonstrate its unsatisfiability.

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

At each stage i , an initial subset $G_i = \{\gamma_1, \dots, \gamma_i\}$ will have been listed by the enumeration procedure. Before moving on to produce γ_{i+1} , apply the procedure from Theorem 57 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 satisfiable is negatively decidable (if it is not, our procedure will tell us that), but not positively decidable (even if it is, our procedure and any similar procedure will not halt in its search for an unsatisfiable subset).

Exercises

1. (a) Show that $(\phi \wedge \psi) \wedge \chi$ is logically equivalent to $\models (\phi \wedge (\psi \wedge \chi))$.
 (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 $c_{\mathcal{A}}$ 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 59 (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 45 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 \mathcal{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 \perp that expresses the constant 0-place function $f : \emptyset \mapsto F$. 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 $\{\rightarrow, \rightarrow^*\}$. 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 $\mathcal{L}_{\neg, \wedge, \vee}$. We say that a sentence ϕ is *dualable* iff for all \mathcal{A} , $(\llbracket \phi \rrbracket_{\mathcal{A}^*})^* = \llbracket \phi^* \rrbracket_{\mathcal{A}}$.
 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 $\mathcal{L}_{\neg, \wedge, \vee}$ are dualable.
8. Let $\mathcal{L}_{\uparrow, \downarrow}$ be the language which has \uparrow and \downarrow as its only connectives.
 (a) For any ϕ in $\mathcal{L}_{\uparrow, \downarrow}$, and any structure \mathcal{A} , show that $\llbracket \phi \rrbracket_{\mathcal{A}} = \mathbf{n}(\llbracket \phi^* \rrbracket_{\mathcal{A}^*})$.

- (b) For any $\mathcal{L}_{\uparrow, \downarrow}$ -sentence ϕ , let ϕ^* be the sentence which results from replacing every connective in ϕ by its dual, and let $\bar{\phi}$ be the result of substituting $(P_i \uparrow P_i)$ for every sentence letter P_i occurring in ϕ .
- Prove that for every $\mathcal{L}_{\uparrow, \downarrow}$ sentence ϕ , ϕ^* is logically equivalent to $(\bar{\phi} \uparrow \bar{\phi})$.
 - Prove that if ϕ and ψ are $\mathcal{L}_{\uparrow, \downarrow}$ sentences, $\phi \models \psi$ iff $\psi^* \models \phi^*$.
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).
- $((Q \vee P) \rightarrow R) \models ((P_1 \wedge \neg R) \rightarrow \neg Q)$;
 - $(\neg(P \vee Q) \wedge (P \leftrightarrow R)) \models ((R \rightarrow P) \wedge \neg(P_1 \wedge R))$;
 - $((Q_2 \leftrightarrow Q) \wedge \neg((R \rightarrow \neg P_1) \vee \neg(P \rightarrow Q))) \models (R_1 \rightarrow (P_2 \rightarrow (\neg P \vee Q)))$;
10. (a) Let Γ be a possibly infinite set of sentences of \mathcal{L}_1 such that $\Gamma \models$. Show that there is a finite disjunction, δ , each disjunct of which is the negation of a sentence in Γ , and such that $\models \delta$.
- (b) Consider the following relation holding between sets of sentences:
- Where Γ and Δ are any sets of sentences, $\Gamma \models^* \Delta$ 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 $\Gamma \models^* \Delta$ and neither is empty (though either or both may be infinite), there is a finite conjunction of \mathcal{L}_1 sentences in Γ ,
- $$\Phi = (\phi_1 \wedge \dots \wedge \phi_m),$$
- and a finite disjunction of \mathcal{L}_1 sentences in Δ ,
- $$\Psi = (\psi_1 \vee \dots \vee \psi_n),$$
- such that $\Phi \models \Psi$. (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 \mathcal{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 \mathcal{L}_1 sentences are logically equivalent.

Answers to selected exercises on page 210.

Chapter 6

Natural Deduction Proofs in \mathcal{L}_1 ; Soundness and Completeness

6.1 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 \mathcal{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 \mathcal{L}_1 ; and thus that we can establish various claims in \mathcal{L}_1 without needing to consider all the \mathcal{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 $\gamma_1, \dots, \gamma_n$, we write this *syntactic sequent*:

$$\gamma_1, \dots, \gamma_n \vdash \phi.$$

This is to be read ' ϕ is *provable from* $\gamma_1, \dots, \gamma_n$ ', or ' $\gamma_1, \dots, \gamma_n$ *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 $\gamma \rightarrow \phi$, *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* occurrence of a sentence – even if there are none!

We can consider two types of obvious rules: those which construct a

$\frac{\begin{array}{c} [\gamma] \\ \vdots \\ \phi \end{array}}{\gamma \rightarrow \phi} \rightarrow\text{Intro}$	$\frac{\gamma \rightarrow \phi \quad \gamma}{\phi} \rightarrow\text{Elim}$
$\frac{\begin{array}{cc} [\phi] & [\phi] \\ \vdots & \vdots \\ \psi & \neg\psi \end{array}}{\neg\phi} \neg\text{Intro}$	$\frac{\begin{array}{cc} [\neg\phi] & [\neg\phi] \\ \vdots & \vdots \\ \psi & \neg\psi \end{array}}{\phi} \neg\text{Elim}$

 Table 6.1: Natural Deduction Rules for $\mathcal{L}_{\neg, \rightarrow}$

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 $\mathcal{L}_{\neg, \rightarrow}$ Let us consider for the moment the fragment of \mathcal{L}_1 involving only \neg and \rightarrow , $\mathcal{L}_{\neg, \rightarrow}$. We have in this fragment four rules, laid out in Table 6.1, 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, $ND_{\neg, \rightarrow}$.

Deduction Theorem for $ND_{\neg, \rightarrow}$

Theorem 6o (Deduction Theorem)

$\Gamma \vdash \psi \rightarrow \phi$ iff $\Gamma, \psi \vdash \phi$.

Suppose $\Gamma, \psi \vdash \phi$. Then we can apply $\rightarrow\text{Intro}$ as follows:

$$\frac{\begin{array}{c} \Gamma, [\psi] \\ \vdots \\ \phi \end{array}}{\psi \rightarrow \phi} \rightarrow\text{Intro}$$

Therefore, $\Gamma \vdash \psi \rightarrow \phi$. Similarly, suppose $\Gamma \vdash \psi \rightarrow \phi$:

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \psi \rightarrow \phi \end{array} \quad \psi}{\phi} \rightarrow\text{Elim}$$

That is, $\Gamma, \psi \vdash \phi$.

$\neg\text{Elim}$ can be replaced by $\neg\neg\text{Elim}$ Consider the rule:

$$\frac{\neg\neg\phi}{\phi} \neg\neg\text{Elim}$$

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:

$$\frac{\neg\neg\phi \quad [\neg\phi]}{\phi} \neg\text{Elim}$$

But with $\neg\neg\text{Elim}$ we can show the rule $\neg\text{Elim}$ is redundant, using $\neg\text{Intro}$:

$$\frac{\begin{array}{c} [\neg\phi] \\ \vdots \\ \psi \end{array} \quad \begin{array}{c} [\neg\phi] \\ \vdots \\ \neg\psi \end{array}}{\neg\neg\phi} \neg\text{Intro}$$

$$\frac{\neg\neg\phi}{\phi} \neg\neg\text{Elim}$$

6.2 Soundness

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 $\mathcal{L}_{\neg, \rightarrow}$. Given the expressive adequacy of $\mathcal{L}_{\neg, \rightarrow}$, that also shows the completeness of the natural deduction system with respect to \mathcal{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 61

The Rules of $ND_{\neg, \rightarrow}$ are sound.

Proof. Suppose that $\Gamma \vdash \phi$. We show by induction on the complexity of proofs that $\Gamma \models \phi$.

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 $\phi \models \phi$, i.e., $\Gamma \models \phi$.

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:

\rightarrow **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, $\Delta \models \psi$ and $\Theta \models \psi \rightarrow \phi$, 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 $\Gamma \models \phi$.

\rightarrow **Intro** From a proof of ϕ on assumptions $\Gamma \cup \{\psi\}$, apply \rightarrow Intro to obtain a proof of $\psi \rightarrow \phi$ on assumption Γ (discharging ψ). But since by hypothesis $\Gamma, \psi \models \phi$, by the deduction theorem, $\Gamma \models \psi \rightarrow \phi$.

\neg 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 $\Gamma, \neg\phi \models \psi$ and $\Gamma, \neg\phi \models \neg\psi$, it follows that $\Gamma, \neg\phi \models$; therefore $\Gamma \models \phi$.

\neg 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 $\Gamma, \phi \models \psi$ and $\Gamma, \phi \models \neg\psi$, it follows that $\Gamma, \phi \models$; therefore $\Gamma \models \neg\phi$.

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. \square

6.3 Completeness

Consistency A set of sentences Γ is called (*syntactically*) *consistent* iff $\Gamma \not\vdash \neg(\phi \rightarrow \phi)$. Otherwise it is *inconsistent*, also written $\Gamma \vdash$. Equivalently, consider this proof:

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \neg(\phi \rightarrow \phi) \end{array} \quad \frac{[\phi] \quad \phi \rightarrow \phi}{\phi \rightarrow \phi} \rightarrow \text{Intro}}{\psi} \neg\text{Elim}$$

If we have a proof of $\neg(\phi \rightarrow \phi)$, 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 62 (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, $\Gamma, \phi \vdash \neg(\phi \rightarrow \phi)$. Using \rightarrow Intro and discharging ϕ , we see that $\Gamma \vdash \phi \rightarrow (\neg(\phi \rightarrow \phi))$. But now use \rightarrow Elim, and obtain $\Gamma \vdash \neg(\phi \rightarrow \phi)$ i.e., Γ is inconsistent. \square

Properties of Maximal Consistent Sets: Negation Completeness
Lemma 63 (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 $\neg(\phi \rightarrow \phi)$ by an application of \neg Elim, showing Γ inconsistent.

R to L: If $\phi \notin \Gamma$, then $\Gamma, \phi \vdash \neg(\phi \rightarrow \phi)$:

$$\frac{\begin{array}{c} \Gamma, [\phi] \\ \vdots \\ \neg(\phi \rightarrow \phi) \end{array} \quad \frac{[\phi]}{\phi \rightarrow \phi} \rightarrow \text{Intro}}{\neg\phi} \neg\text{Intro}$$

Therefore, $\Gamma \vdash \neg\phi$; by Deductive Closure Lemma, $\neg\phi \in \Gamma$. \square

Properties of Maximal Consistent Sets: Conditional Completeness
Lemma 64 (Conditional Completeness)

If Γ is any maximal consistent set, then $\phi \rightarrow \psi \in \Gamma$ iff whenever $\phi \in \Gamma$, it follows that $\psi \in \Gamma$.

Proof. L to R: Suppose $\phi \rightarrow \psi \in \Gamma$. Then $\Gamma \vdash \phi \rightarrow \psi$; and if $\phi \in \Gamma$, then $\Gamma \vdash \phi$. Apply \rightarrow 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 \rightarrow Intro and discharge ϕ to show $\Gamma \vdash \phi \rightarrow \psi$.

2. $\phi \notin \Gamma$. Then $\Gamma, \phi \vdash \neg(\phi \rightarrow \phi)$. But then $\Gamma, \phi \vdash \psi$ (by a similar proof as just above); applying \rightarrow Intro and discharge ϕ , $\Gamma \vdash \phi \rightarrow \psi$. \square

Properties of Maximal Consistent Sets: Satisfiability

Theorem 65 (Maximal Consistent Sets Satisfiable)

Every maximal consistent set is satisfiable (i.e., has a model).

Proof. Suppose Γ is a maximal consistent set in $\mathcal{L}_{\rightarrow, \neg}$. For s a sentence letter, define a structure \mathcal{A} : $\mathcal{A}(s) = T$ iff $s \in \Gamma$.

For any ϕ , $\llbracket \phi \rrbracket_{\mathcal{A}} = T$ 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, $\llbracket \psi \rrbracket_{\mathcal{A}} = F$; iff by the clause on \neg , $\llbracket \phi \rrbracket_{\mathcal{A}} = T$.
2. $\phi = (\psi \rightarrow \chi)$. $\phi \in \Gamma$ iff if $\psi \in \Gamma$ then $\chi \in \Gamma$ (by Conditional Completeness); iff if $\llbracket \psi \rrbracket_{\mathcal{A}} = T$ then $\llbracket \chi \rrbracket_{\mathcal{A}} = T$; iff $\llbracket \phi \rrbracket_{\mathcal{A}} = T$. \square

Completeness

Theorem 66 (Completeness for $\mathcal{L}_{\neg, \rightarrow}$)

If $\Gamma \models \phi$, then $\Gamma \vdash \phi$.

Proof. We prove the *contrapositive*. So assume that $\Gamma \not\vdash \phi$. (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 $\Gamma^+ \not\vdash \phi$. 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, \mathcal{A} . Since $\Gamma \subseteq \Gamma^+$, for all $\gamma \in \Gamma$, $\llbracket \gamma \rrbracket_{\mathcal{A}} = T$; but since $\neg\phi \in \Gamma^+$, $\llbracket \neg\phi \rrbracket_{\mathcal{A}} = T$, so $\llbracket \phi \rrbracket_{\mathcal{A}} = F$. So $\Gamma \not\models \phi$, as required.

We now show how to construct Γ^+ from Γ , completing the proof. \square

The Construction of Maximal Consistent Γ^+ Supposing $\Gamma \not\vdash \phi$, construct Γ^+ as follows.

1. Enumerate the sentences of $\mathcal{L}_{\neg, \rightarrow}$, $\{\sigma_1, \sigma_2, \dots\}$.
2. Let $\Gamma_0 = \Gamma$.
- 3.

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\neg\sigma_{n+1}\} & \text{if } \Gamma_n, \sigma_{n+1} \vdash \phi; \\ \Gamma_n \cup \{\sigma_{n+1}\} & \text{otherwise (i.e., if } \Gamma_n, \sigma_{n+1} \not\vdash \phi). \end{cases}$$

4. Let $\Gamma^+ = \bigcup_i \Gamma_i$.

It is clear that $\Gamma \subseteq \Gamma^+$. It is also clear that $\Gamma^+ \not\vdash \phi$, 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 \mathcal{L}_1 We've shown the Completeness Theorem for the restricted language $\mathcal{L}_{\neg, \rightarrow}$. But by Expressive Adequacy, any sentence of \mathcal{L}_1 can be expressed in $\mathcal{L}_{\neg, \rightarrow}$. So we can assure ourselves that any truth function expressible in \mathcal{L}_1 can be expressed in $\mathcal{L}_{\neg, \rightarrow}$, and any valid argument in \mathcal{L}_1 has a corresponding provably correct sequent in $\mathcal{L}_{\neg, \rightarrow}$.

For completeness of \mathcal{L}_1 , we cannot rest there. What if the rules of \mathcal{L}_1 don't suffice to show the equivalence of some arbitrary \mathcal{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 $\Gamma \models$ 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 $\Gamma' \models$ where Γ' is a finite subset of Γ .

6.4 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 \neg Elim in the presence of $\neg\neg$ Elim, and vice versa.

The Language \mathcal{L}_{DP} Consider the language \mathcal{L}_{DP} which contains all sentence letters, and all sentences involving only the connectives \wedge, \rightarrow , and also contains a *sentential constant* \perp . 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 \mathcal{A} , $|\perp|_{\mathcal{A}} = F$. The atomic sentences of \mathcal{L}_{DP} thus include \perp and all sentence letters; complex sentences are then built up in the usual way.

$\frac{\begin{array}{c} [\phi] \\ \vdots \\ \psi \end{array} \quad \begin{array}{c} [\phi] \\ \vdots \\ \psi \rightarrow \perp \end{array}}{\phi \rightarrow \perp} \neg\text{Intro}_{DP}$	$\frac{\begin{array}{c} [\phi \rightarrow \perp] \\ \vdots \\ \psi \end{array} \quad \begin{array}{c} [\phi \rightarrow \perp] \\ \vdots \\ \psi \rightarrow \perp \end{array}}{\phi} \neg\text{Elim}_{DP}$
--	---

 Table 6.2: Negation rules in \mathcal{L}_{DP}

Translating between \mathcal{L}_{DP} and \mathcal{L}_1 It's clear that the \mathcal{L}_1 sentence ' $\neg\phi$ ' expresses the same truth function as the \mathcal{L}_{DP} sentence ' $\phi \rightarrow \perp$ ' (consideration of a truth table will suffice to show this). Using this translation, we can give \mathcal{L}_{DP} -versions of the natural deduction rules for \neg in \mathcal{L}_1 , as seen in Table 6.2.

With the negation rules as specified, we can prove this:

Theorem 67 (Alternative Negation Rules for \mathcal{L}_{DP})

Π is a proof of some sequent in the language \mathcal{L}_{DP} that makes a use u of one of the Table 6.2 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 \rightarrow):

$$\frac{\begin{array}{c} [\phi \rightarrow \perp] \\ \vdots \\ \perp \end{array}}{\phi} \perp_C$$

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 :

$$\frac{\frac{\begin{array}{c} [\phi] \\ \vdots \\ \psi \end{array} \quad \begin{array}{c} [\phi] \\ \vdots \\ \psi \rightarrow \perp \end{array}}{\perp} \rightarrow \text{Elim}}{\phi \rightarrow \perp} \rightarrow \text{Intro}$$

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 \perp_C and the rule $\rightarrow\text{Elim}$:

$$\begin{array}{c}
 [\phi \rightarrow \perp] \quad [\phi \rightarrow \perp] \\
 \vdots \quad \quad \quad \vdots \\
 \psi \quad \quad \quad \psi \rightarrow \perp \\
 \hline
 \perp \rightarrow \text{Elim} \\
 \perp \\
 \hline
 \phi \perp_C
 \end{array}$$

Only if: Suppose Π is a proof which makes use of the new rule \perp_C . Then we may replace that use by the following construction which uses only the old rules for \rightarrow and $\neg\text{Elim}_{DP}$.

$$\begin{array}{c}
 [\phi \rightarrow \perp] \\
 \vdots \\
 \perp \\
 \hline
 \phi \\
 \hline
 \perp \rightarrow \perp \rightarrow \text{Intro} \\
 \neg\text{Elim}_{DP}
 \end{array}$$

□

Reducing Complexity of Proofs Return now to the \mathcal{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 68 (Reducing Complexity of Proofs)

If Π is a proof in Halbach's system for $\mathcal{L}_{\rightarrow, \wedge, \neg}$ that $\Gamma \vdash \phi$, in which the most complicated sentence resulting from an application of $\neg\text{Elim}$ 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 $\neg\text{Elim}$ are less complicated than ψ . Indeed, there exists a proof Π^\dagger of that sequent, in which the result of every application of $\neg\text{Elim}$ is atomic.

Proof. Suppose that the most complicated result of \neg Elim in Π is ψ . So part of Π looks like this:

$$\frac{\begin{array}{c} \Gamma \quad [\neg\psi] \quad \Delta \quad [\neg\psi] \\ \vdots \\ \chi \end{array} \quad \begin{array}{c} \vdots \\ \neg\chi \end{array}}{\psi} \neg\text{Elim}$$

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:

$$\frac{\begin{array}{c} \Gamma \quad \frac{[\pi] \quad [\neg\pi]}{\neg\neg\pi} \neg\text{Intro} \\ \vdots \\ \chi \end{array} \quad \begin{array}{c} \Delta \quad \frac{[\pi] \quad [\neg\pi]}{\neg\neg\pi} \neg\text{Intro} \\ \vdots \\ \neg\chi \end{array}}{\neg\pi} \neg\text{Intro}$$

2. $\psi = \pi \wedge \xi$. Proof left for Exercise.

3. $\psi = \pi \rightarrow \xi$. In which case replace that part of the proof sketched above with this one:

$$\frac{\begin{array}{c} \Gamma \quad \frac{\frac{[\pi \rightarrow \xi] \quad [\pi]}{\xi} \rightarrow \text{Elim} \quad [\neg\xi]}{\neg(\pi \rightarrow \xi)} \neg\text{Intro} \\ \vdots \\ \chi \end{array} \quad \begin{array}{c} \Delta \quad \frac{\frac{[\pi \rightarrow \xi] \quad [\pi]}{\xi} \rightarrow \text{Elim} \quad [\neg\xi]}{\neg(\pi \rightarrow \xi)} \neg\text{Intro} \\ \vdots \\ \neg\chi \end{array}}{\frac{\xi}{\pi \rightarrow \xi} \rightarrow \text{Intro}} \neg\text{Elim}$$

Notice that applying these replacements repeatedly to uses of \neg Elim in a proof will eventually replace all complex sentences in uses of \neg Elim

by simpler ones. All sentences of \mathcal{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 \neg Elim. \square

It is clear that we can put the results of Theorems 67 and 68 together, to establish that, in the system which involves the normal rules for \wedge and \rightarrow , and has \perp_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, Π^\dagger , in which the result of any application of \perp_C is atomic. (I leave the proof as a problem.)

6.5 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 \mathbb{L} is very simple. It has three *axiom-schemata*, in the language $\mathcal{L}_{\neg, \rightarrow}$:

A1 $\vdash_{\mathbb{L}} (\phi \rightarrow (\psi \rightarrow \phi))$;

A2 $\vdash_{\mathbb{L}} ((\phi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi)))$;

A3 $\vdash_{\mathbb{L}} ((\neg\psi \rightarrow \neg\phi) \rightarrow (\phi \rightarrow \psi))$.

Any sentence of $\mathcal{L}_{\neg, \rightarrow}$ 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 $\vdash_{\mathbb{L}} \phi$ and $\vdash_{\mathbb{L}} \phi \rightarrow \psi$, then $\vdash_{\mathbb{L}} \psi$.

A *proof* in \mathbb{L} is any sequence of theorems, each of which is either an axiom or follows by *modus ponens* from earlier theorems.

A Proof in \mathbb{L} The system \mathbb{L} 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. $\vdash_{\mathbb{L}} (P \rightarrow ((P \rightarrow P) \rightarrow P)) \rightarrow ((P \rightarrow (P \rightarrow P)) \rightarrow (P \rightarrow P))$ (A2)
2. $\vdash_{\mathbb{L}} (P \rightarrow ((P \rightarrow P) \rightarrow P))$ (A1)
3. $\vdash_{\mathbb{L}} ((P \rightarrow (P \rightarrow P)) \rightarrow (P \rightarrow P))$ (MP 1, 2)
4. $\vdash_{\mathbb{L}} (P \rightarrow (P \rightarrow P))$ (A1)
5. $\vdash_{\mathbb{L}} (P \rightarrow P)$ (MP 3, 4)

This is (apparently) the *shortest* proof of $P \rightarrow P$ in \mathbb{L} !

Soundness of \mathbb{L} It is a trivial exercise (and hence left for an exercise) to show that \mathbb{L} is sound. That is, show:

- All the axioms of \mathbb{L} are tautologies.
- The rule of \mathbb{L} preserves tautologousness.

Equivalence of \mathbb{L} and $ND_{\neg, \rightarrow}$ We now show that, perhaps surprisingly, the proof systems \mathbb{L} and $ND_{\neg, \rightarrow}$ are equivalent – everything that can be proved in one can be proved in the other. This shows, too, that \mathbb{L} is sound and complete.

Everything provable in \mathbf{L} is provable in ND

Theorem 69

If $\vdash_{\mathbf{L}} \phi$ 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:

$$\frac{\frac{[\phi]}{\psi \rightarrow \phi} \rightarrow \text{Intro}}{\phi \rightarrow (\psi \rightarrow \phi)} \rightarrow \text{Intro}$$

□

Proofs from Assumptions in \mathbf{L} The ugly proof of $\vdash_{\mathbf{L}} P \rightarrow P$ two paragraphs back shows the benefits of working with *assumptions*. A proof of ϕ from assumptions Γ in \mathbf{L} 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 70 (Deduction Theorem for \mathbf{L})

$\Gamma \vdash_{\mathbf{L}} \phi$ iff for some finite set of sentences $\gamma_1, \dots, \gamma_n$ actually used in the proof of ϕ , $\vdash_{\mathbf{L}} (\gamma_1 \rightarrow (\gamma_2 \rightarrow \dots (\gamma_n \rightarrow \phi) \dots))$.

It suffices to show that $\Gamma, \psi \vdash_{\mathbf{L}} \phi$ iff $\Gamma \vdash_{\mathbf{L}} \psi \rightarrow \phi$. I leave the proof for an exercise.

Everything Provable in ND is provable in \mathbf{L}

Theorem 71

If $\vdash \phi$, then $\vdash_{\mathbf{L}} \phi$.

Proof. Induction on length of proofs: we show that the shortest proofs of ND are provable in \mathbb{L} , and that the rules are provable transitions.

Certainly the shortest proof, the single node ϕ , shows that $\phi \vdash \phi$; but $\phi \vdash_{\mathbb{L}} \phi$ is shown by the deduction theorem given $\vdash_{\mathbb{L}} \phi \rightarrow \phi$.

Suppose we extend some existing proofs by applying the rules of ND :

\rightarrow **Elim** We have $\Gamma \vdash \phi$ and $\Gamma \vdash \phi \rightarrow \psi$, and extend by \rightarrow Elim. But this is obviously *modus ponens*, so $\Gamma \vdash_{\mathbb{L}} \psi$.

The case of \rightarrow Intro is shown by the deduction theorem. I leave you to show the cases for negation. \square

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 67 and 68, 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 Ślupecki (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 \mathbb{L} at pp. 217–9.

Exercises

1. (a) Devise introduction and elimination rules for a natural deduction system for the language \mathcal{L}_1 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 \mathcal{L}_1 we replaced the \neg Elim rule by the following, *ex falso quodlibet*:

$$\frac{\psi \quad \neg\psi}{\phi} \text{ EFQ}$$

- 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 \neg Intro by the following rule, *tertium non datur*, yielding the system ND'' ?

$$\frac{\begin{array}{c} [\phi] \\ \vdots \\ \psi \end{array} \quad \begin{array}{c} [\neg\phi] \\ \vdots \\ \psi \end{array}}{\psi} \text{TND}$$

- (c) Consider the natural deduction system $ND_{\rightarrow, \vee}$ for the language $\mathcal{L}_{\rightarrow, \vee}$ in which the only connectives are \rightarrow, \vee . Suppose we introduce to the language the zero-place sentential constant \perp , and add the following rules to the proof system, yielding $ND_{\rightarrow, \vee, \perp}$:

$$\frac{\perp}{\phi} \perp \quad \frac{\begin{array}{c} [\phi \rightarrow \perp] \\ \vdots \\ \perp \end{array}}{\phi} \text{C}$$

- i. Give a natural translation between $\mathcal{L}_{\rightarrow, \vee, \perp}$ and $\mathcal{L}_{\rightarrow, \vee, \neg}$.
 - ii. Using that translation, can you show that the proof system $ND_{\rightarrow, \vee, \perp}$ just introduced is equivalent to the natural deduction system $ND_{\rightarrow, \vee, \neg}$ involving just Halbach's rules for \rightarrow, \vee and \neg ?
- (d) A mystery connective \oplus has the following introduction and elimination rules:

$$\frac{\phi \oplus \psi}{\psi} \oplus\text{Elim} \quad \frac{\phi}{\phi \oplus \psi} \oplus\text{Intro}$$

- 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 $((P \rightarrow Q) \rightarrow P) \rightarrow P$ is a tautology.
 - ii. Give a natural deduction proof of $((P \rightarrow Q) \rightarrow P) \rightarrow P$.
 - 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. Using the model soundness proof for the restricted natural deduction system $ND_{\neg, \rightarrow}$, show, by analysing proofs constructed using the remaining rules of the full Halbach system ND , that the system ND is sound.
 3. The Completeness proof for $\mathcal{L}_{\neg, \rightarrow}$ relied on various lemmas concerning the behaviour of maximal consistent sets with regard to the connectives \neg and \rightarrow . Show that
 - (a) analogous lemmas hold for \wedge , \vee and \leftrightarrow .
 - (b) the Satisfiability Theorem holds for the full language \mathcal{L}_1 .
 - (c) the full Natural Deduction system is complete for \mathcal{L}_1 with respect to the intended semantics.
 4.
 - (a) Show that the axiomatic system \mathbb{L} introduced is sound.
 - (b) Show that the axioms A2 and A3 of \mathbb{L} are provable in natural deduction:
 - i. A2. $\vdash_{\mathbb{L}} ((\phi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi)))$;
 - ii. A3. $\vdash_{\mathbb{L}} ((\neg\psi \rightarrow \neg\phi) \rightarrow (\phi \rightarrow \psi))$.
 5. Suppose that $\Gamma, \psi \vdash_{\mathbb{L}} \phi$, shown by a proof Π . Define the ψ -transform of Π , written Π^ψ , as the sequence that results from prefixing ' $\psi \rightarrow$ ' to the front of every sentence in Π . Show that each sentence in Π^ψ is provable from assumptions Γ in \mathbb{L} . Since obviously ψ appears on Π , $\psi \rightarrow \psi$ appears on Π^ψ , and that is provable from Γ . So you need to show that

- (a) the prefixed axioms (i.e., those $\psi \rightarrow \delta$ in Π^ψ where δ is an axiom) are provable from Γ (hint: use A1);
- (b) the prefixed assumptions (i.e., those $\psi \rightarrow \gamma$ in Π^ψ where $\gamma \in \Gamma$) are provable from Γ ;
- (c) If χ followed by *modus ponens* from early claims in Π , $\psi \rightarrow \chi$ follows from earlier prefixed claims in Π^ψ (hint: use A2).

Using these results, show the deduction theorem for \mathbb{L} .

6. Show the other cases for the equivalence of ND and \mathbb{L} :¹

- (a) Show that if $\Gamma, \phi \vdash_{\mathbb{L}} \psi$ and $\Gamma, \phi \vdash_{\mathbb{L}} \neg\psi$, then $\Gamma \vdash_{\mathbb{L}} \neg\phi$.
- (b) Show that if $\Gamma, \neg\phi \vdash_{\mathbb{L}} \psi$ and $\Gamma, \neg\phi \vdash_{\mathbb{L}} \neg\psi$, then $\Gamma \vdash_{\mathbb{L}} \phi$. (Equivalently, show that if $\Gamma \vdash_{\mathbb{L}} \neg\neg\phi$, then $\Gamma \vdash_{\mathbb{L}} \phi$.)

7. A fellow student argues as follows:

If the Natural Deduction system introduced by Halbach is complete, then whenever $\Gamma \models \phi$, there is a proof 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?

- 8. Prove the omitted case (where the complex sentence is a conjunction) in the proof of Theorem 68.
- 9. Show that, in the system which involves the normal rules for \wedge and \rightarrow , and has \perp_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, Π^\dagger , in which the result of any application of \perp_C is atomic.

¹These proofs are quite elusive; I propose treating this question as optional.

Chapter 7

The Syntax and Semantics of \mathcal{L}_2

7.1 Syntax of \mathcal{L}_2

\mathcal{L}_2 We now have a good grip on \mathcal{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 \mathcal{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 \mathcal{L}_1 to capture valid arguments is that the analysis \mathcal{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 \mathcal{L}_1 formalisation is the incorrect sequent: $P \models Q$. 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 \mathcal{L}_2 : \mathcal{L}_1 alphabet and Predicate Letters The alphabet of \mathcal{L}_2 – the constituents out of which \mathcal{L}_2 -sentences may be formed – includes the alphabet of \mathcal{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 \mathcal{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 $n \geq 0$, we introduce infinitely many predicate letters: $P^n, Q^n, R^n, P_1^n, Q_1^n, R_1^n, \dots$. Where $n = 0$, we omit the index. (So any expression of the form P_k^n, Q_k^n , or R_k^n , 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 \mathcal{L}_1 .

The Syntax of \mathcal{L}_2 : Constants, Variables, Quantifiers The alphabet of \mathcal{L}_2 includes infinitely many *constants*, $a, b, c, a_1, b_1, c_1, \dots$.

\mathcal{L}_2 also contains infinitely many *variables*, $x, y, z, x_1, y_1, z_1, \dots$.

And \mathcal{L}_2 contains infinitely many *quantifiers*, $\forall v$ and $\exists v$, where v 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 \mathcal{L}_2

Definition 85 (Atomic Formula). If Φ^n is a predicate letter of arity $n \geq 0$, and τ_1, \dots, τ_n are n terms (not necessarily all distinct), then $\Phi\tau_1 \dots \tau_n$ is an *atomic formula* of \mathcal{L}_2 .

It follows that all sentence letters are atomic formulae.

Definition 86 (Formula).

1. Any atomic formula is a formula of \mathcal{L}_2 .
2. If ϕ and ψ are formulae of \mathcal{L}_2 , then so are $\neg\phi$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$.
3. If $\forall v$ and $\exists v$ are quantifiers and ϕ is a formula, then so are $\forall v\phi$ and $\exists v\phi$.
4. Nothing else is a formula of \mathcal{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 87 (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 88 (Sentence of \mathcal{L}_2). An \mathcal{L}_2 -formula ϕ is a *sentence* (or ‘closed formula’) of \mathcal{L}_2 iff no variable occurs freely in ϕ .

7.2 Semantics of \mathcal{L}_2

Semantics for \mathcal{L}_2 : Domains As in \mathcal{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 \mathcal{L}_1 structures cannot suffice for \mathcal{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 $D \underbrace{\times \dots \times}_n D = D^n$.

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 $X = \{\langle x_1 \rangle, \dots, \langle x_n \rangle, \dots\}$, rather than $X' = \{x_1, \dots, x_n, \dots\}$? 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.

\mathcal{L}_2 Structures and Variable Assignments

Definition 89 (\mathcal{L}_2 -Structure). An \mathcal{L}_2 -structure \mathcal{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 90 (Variable Assignment). A *variable assignment* α over \mathcal{A} is a function which assigns each variable in \mathcal{L}_2 a member of $D_{\mathcal{A}}$.

Semantic Values of Simple Expressions Let us extend the notation $\llbracket \cdot \rrbracket_{\mathcal{A}}$ to mean the semantic value of any \mathcal{L}_2 expression in structure \mathcal{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 \mathcal{A} and variable assignment α , $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = I_{\mathcal{A}}(\phi)$ for all ϕ in the domain of $I_{\mathcal{A}}$. (All constants, and predicate letters, including sentence letters, thus receive a semantic value at this stage.) The variables of \mathcal{L}_2 get the obvious interpretation: $\llbracket v \rrbracket_{\mathcal{A}}^{\alpha} = \alpha(v)$. We now assign truth values to the formulae of \mathcal{L}_2 .

Satisfaction

Definition 91 (Satisfaction). Supposing ϕ, ψ to be formulae of \mathcal{L}_2 , v a variable, \mathcal{A} a \mathcal{L}_2 -structure, and α a variable assignment over \mathcal{A} . We say that α *satisfies* ϕ under \mathcal{A} , $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$, as follows:

- $\llbracket \Phi \tau_1 \dots \tau_n \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\langle \llbracket \tau_1 \rrbracket_{\mathcal{A}}^{\alpha}, \dots, \llbracket \tau_n \rrbracket_{\mathcal{A}}^{\alpha} \rangle \in \llbracket \Phi \rrbracket_{\mathcal{A}}^{\alpha}$, where Φ is a n -ary predicate letter ($n > 0$), and each τ_i is a term.¹
- $\llbracket \neg \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = F$.
- $\llbracket \phi \wedge \psi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \psi \rrbracket_{\mathcal{A}}^{\alpha} = T$.
- $\llbracket \phi \vee \psi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$ or $\llbracket \psi \rrbracket_{\mathcal{A}}^{\alpha} = T$.
- $\llbracket \phi \rightarrow \psi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = F$ or $\llbracket \psi \rrbracket_{\mathcal{A}}^{\alpha} = T$.
- $\llbracket \phi \leftrightarrow \psi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \psi \rrbracket_{\mathcal{A}}^{\alpha}$.
- $\llbracket \forall v \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\beta} = T$ for all variable assignments β over \mathcal{A} differing from α *at most* in their assignment to v .
- $\llbracket \exists v \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\beta} = T$ for at least one variable assignment β over \mathcal{A} differing from α *at most* in their assignment to v .

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 $\forall v \phi$ is satisfied just in case ϕ is satisfied in every variable assignment that holds everything fixed except possibly the semantic value of v – i.e., no matter what v denotes, ϕ is satisfied.

¹Recall the discussion of ordered 1-tuples at the beginning of this section when applying this definition to monadic predicates.

- An existentially quantified sentence $\exists v\phi$ is satisfied just in case ϕ is satisfied in at least one variable assignment that holds everything fixed except possibly the semantic value of v – i.e., there is a semantic value for v on which ϕ is satisfied.

Truth The notion of truth is then defined in terms of satisfaction:

Definition 92 (Truth). A sentence ϕ is *true* in \mathcal{A} iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$ for every variable assignment α over \mathcal{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 any constant is substituted for x , not unconditionally as a sentence does.)

7.3 Some Semantic Theorems About \mathcal{L}_2

Semantic Sequents for \mathcal{L}_2 We now extend the semantic turnstile \models to \mathcal{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 \mathcal{L}_2 -structures, ϕ is a *tautology* or *logical truth*, written $\models \phi$.
2. If ϕ is true in no \mathcal{L}_2 -structure, ϕ is a *contradiction*, written $\phi \models$.
3. A set Γ of \mathcal{L}_2 sentences is *semantically consistent* iff there is some \mathcal{L}_2 -structure in which each $\gamma \in \Gamma$ is true; otherwise Γ is inconsistent, written $\Gamma \models$.
4. ϕ is *logically equivalent* to ψ iff both sentences are true in the same \mathcal{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 93 (Validity in \mathcal{L}_2). An argument from Γ to ϕ is *valid* (Γ entails ϕ) iff in every \mathcal{L}_2 -structure in which each $\gamma \in \Gamma$ is true, ϕ is true also. We write again $\Gamma \models \phi$.

Some Theorems About Entailment in \mathcal{L}_2 Many of the theorems about \models we proved for \mathcal{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 $\Gamma, \phi \models \psi$ and $\Gamma \models \phi$, then $\Gamma \models \psi$.

Transitivity If $\Gamma \models \phi$ and $\phi \models \psi$, then $\Gamma \models \psi$.

Contraposition If $\phi \models \psi$ iff $\neg\psi \models \neg\phi$.

Deduction Theorem $\Gamma, \phi \models \psi$ iff $\Gamma \models \phi \rightarrow \psi$.

We now prove some theorems that depend on the new machinery we've introduced.

Quantifier Interdefinability

Theorem 72 (Interdefinable Quantifiers)

For all structures \mathcal{A} , and all variable assignments α ,

- $\llbracket \forall v \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \neg \exists v \neg \phi \rrbracket_{\mathcal{A}}^{\alpha};$
- $\llbracket \exists v \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \neg \forall v \neg \phi \rrbracket_{\mathcal{A}}^{\alpha}.$

Proof. I show only the first case. $\llbracket \forall v \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff for all v -variant assignments β , $\llbracket \phi \rrbracket_{\mathcal{A}}^{\beta} = T$; iff $\llbracket \neg \phi \rrbracket_{\mathcal{A}}^{\beta} = F$; iff $\llbracket \exists v \neg \phi \rrbracket_{\mathcal{A}}^{\alpha} = F$ (if not, there would be a β where $\llbracket \neg \phi \rrbracket_{\mathcal{A}}^{\beta} = T$); iff $\llbracket \neg \exists v \neg \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$. \square

Substitution of Co-Designating Terms

Theorem 73 (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 \mathcal{A} and any variable assignment α : If $\llbracket\tau_1\rrbracket_{\mathcal{A}}^{\alpha} = \llbracket\tau_2\rrbracket_{\mathcal{A}}^{\alpha}$, then $\llbracket\phi\rrbracket_{\mathcal{A}}^{\alpha} = \llbracket\phi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha}$.

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 $\llbracket\Phi\theta_1 \dots \theta_n\rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\langle \llbracket\theta_1\rrbracket_{\mathcal{A}}^{\alpha}, \dots, \llbracket\tau_1\rrbracket_{\mathcal{A}}^{\alpha}, \dots, \llbracket\theta_n\rrbracket_{\mathcal{A}}^{\alpha} \rangle \in \llbracket\Phi\rrbracket_{\mathcal{A}}^{\alpha}$; since $\llbracket\tau_1\rrbracket_{\mathcal{A}}^{\alpha} = \llbracket\tau_2\rrbracket_{\mathcal{A}}^{\alpha}$, $\langle \llbracket\theta_1\rrbracket_{\mathcal{A}}^{\alpha}, \dots, \llbracket\tau_2\rrbracket_{\mathcal{A}}^{\alpha}, \dots, \llbracket\theta_n\rrbracket_{\mathcal{A}}^{\alpha} \rangle \in \llbracket\Phi\rrbracket_{\mathcal{A}}^{\alpha}$, i.e., $\llbracket\phi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$.

Induction step: Suppose ϕ is complex, and the theorem holds for ψ , χ less complex than ϕ and *each* variable assignment α over \mathcal{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 $\llbracket\neg\psi\rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\llbracket\neg\psi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$; by the clause on negation, that holds iff $\llbracket(\neg\psi)[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$, i.e., iff $\llbracket\phi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$.
- Suppose $\phi = \psi \vee \chi$. $\llbracket\psi \vee \chi\rrbracket_{\mathcal{A}}^{\alpha} = T$ iff either $\llbracket\psi\rrbracket_{\mathcal{A}}^{\alpha} = T$ or $\llbracket\chi\rrbracket_{\mathcal{A}}^{\alpha} = T$; by the induction hypothesis, iff $\llbracket\psi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$ or $\llbracket\chi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$; by substitution, iff $\llbracket(\psi \vee \chi)[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$, i.e., iff $\llbracket\phi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$.
- Suppose $\phi = \forall v\psi$. $\llbracket\forall v\psi\rrbracket_{\mathcal{A}}^{\alpha} = T$ iff for any variable assignment β which agrees with α except perhaps on v , $\llbracket\psi\rrbracket_{\mathcal{A}}^{\beta} = T$. By the induction hypothesis, and the fact that it holds for all variable assignments, that holds iff $\llbracket\psi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\beta} = T$ (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 $\llbracket\forall v\psi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$ (again, it is required $v \neq \tau_2$), i.e., iff $\llbracket\phi[\tau_2/\tau_1]\rrbracket_{\mathcal{A}}^{\alpha} = T$. □

Satisfaction of Sentences

Theorem 74 (Satisfaction of Sentences)

If ϕ is a \mathcal{L}_2 -sentence, then for every structure \mathcal{A} and all variable assignments α, β , $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \phi \rrbracket_{\mathcal{A}}^{\beta}$.

Proof. by induction on complexity. *Base case:* ϕ is an atomic formula. It is a sentence, so no variables occur in $\phi = \Phi\tau_1 \dots \tau_n$; since constants receive a constant interpretation in \mathcal{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 \mathcal{A} . The only interesting cases are the quantifiers, so let $\phi = \forall v\chi$.

1. χ is a sentence (so the quantification is vacuous). Then for any α, β $\llbracket \chi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \chi \rrbracket_{\mathcal{A}}^{\beta}$, so $\llbracket \forall v\chi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \forall v\chi \rrbracket_{\mathcal{A}}^{\beta}$.
2. χ is not a sentence; since ϕ is, only v occurs free in χ . Suppose for some α, β , $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} \neq \llbracket \phi \rrbracket_{\mathcal{A}}^{\beta}$. Without loss of generality, suppose $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$. Then for every variable assignment γ differing from α at most on v , $\llbracket \chi \rrbracket_{\mathcal{A}}^{\gamma} = T$.

But for at least one variable assignment δ differing from β on at most v , $\llbracket \chi \rrbracket_{\mathcal{A}}^{\delta} = F$.

Let τ not occur in χ , and let $\llbracket \tau \rrbracket_{\mathcal{A}}^{\delta} = \llbracket v \rrbracket_{\mathcal{A}}^{\delta}$. Then by the Substitution of Terms Theorem, $\llbracket \chi \rrbracket_{\mathcal{A}}^{\delta} = \llbracket \chi[\tau/v] \rrbracket_{\mathcal{A}}^{\delta} = F$. By the induction hypothesis, for every γ , $\llbracket \chi[\tau/v] \rrbracket_{\mathcal{A}}^{\gamma} = F$. In particular, for some γ^* which is like α except that it assigns $\llbracket \tau \rrbracket_{\mathcal{A}}$ to be the extension of v , $\llbracket \chi[\tau/v] \rrbracket_{\mathcal{A}}^{\gamma^*} = F$. But since $\llbracket \tau \rrbracket_{\mathcal{A}}^{\gamma^*} = \llbracket v \rrbracket_{\mathcal{A}}^{\gamma^*}$, $\llbracket \chi \rrbracket_{\mathcal{A}}^{\gamma^*} = F$; contradicting our assumption that every v -variant of α satisfied χ . So there is no case where $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} \neq \llbracket \phi \rrbracket_{\mathcal{A}}^{\beta}$. \square

Quantifiers: Universal Elimination/Existential Introduction**Theorem 75**

Suppose ϕ is a formula in which only v occurs free, and τ any constant. Then:

- $\forall v\phi \models \phi[\tau/v]$;
- $\phi[\tau/v] \models \exists v\phi$.

Proof. I only show the second. Suppose that \mathcal{A} is a structure in which $\phi[\tau/v]$ is true. Then in every variable assignment α according to which $\llbracket \tau \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket v \rrbracket_{\mathcal{A}}^{\alpha}$, $\llbracket \phi[\tau/v] \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha}$ (by the substitution of co-designating terms). There will be such a variable assignment α ; so by the clause for \exists , $\llbracket \exists v\phi \rrbracket_{\mathcal{A}} = T$.

Quantifiers: Universal Introduction/Existential Elimination**Theorem 76**

Suppose ϕ is a formula in which at most v occurs free, and τ any constant not occurring in Γ or ϕ . Then:

- *If $\Gamma \models \phi[\tau/v]$, then $\Gamma \models \forall v\phi$.*
- *If $\Gamma, \phi[\tau/v] \models \psi$ then $\Gamma, \exists v\phi \models \psi$, provided τ does not occur in ψ .*

Proof. Suppose that every structure \mathcal{A} which makes Γ true makes $\phi[\tau/v]$ true. τ does not occur in Γ , so every structure \mathcal{A}' just like \mathcal{A} except perhaps in the value of $I_{\mathcal{A}'}(\tau)$ also makes all of Γ true. Then $\phi[\tau/v]$ is true in every such τ -variant structure \mathcal{A}' . Suppose β is a variable assignment where $\llbracket \phi \rrbracket_{\mathcal{A}}^{\beta} = F$. There is some \mathcal{A}' where $I_{\mathcal{A}'}(\tau) = \llbracket v \rrbracket_{\mathcal{A}'}^{\beta}$, so $\llbracket \phi[\tau/v] \rrbracket_{\mathcal{A}'}^{\beta} = F$. Contradiction; there is no such variable assignment, so $\forall v\phi$ is true in every structure \mathcal{A} which makes all of Γ true.

Substitution of Equivalent Formulae

Theorem 77 (Substitution of Equivalent Formulae)

Suppose ϕ and ψ are formulae, in which v_1, \dots, v_n occur free. Suppose δ is a some sentence in which ϕ occurs as a subformula. Then $\forall v_1 \dots \forall v_n (\phi \leftrightarrow \psi) \models \delta \leftrightarrow \delta[\psi/\phi]$.

Proof. In every structure \mathcal{A} in which $\forall v_1 \dots \forall v_n (\phi \leftrightarrow \psi)$ is true, $\phi \leftrightarrow \psi$ is satisfied by every variable assignment agreeing on variables except possibly the v_i s. But clearly if for every variable assignment α $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \psi \rrbracket_{\mathcal{A}}^{\alpha}$, a straightforward induction on complexity of δ will establish the result. \square

7.4 Alternative Semantics for \mathcal{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 \mathcal{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 one 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:
 - $\llbracket \forall \xi \phi \rrbracket_{\mathcal{A}} = T$ iff for every individual $i \in D_{\mathcal{A}}$, and every structure \mathcal{A}' just like \mathcal{A} except perhaps in the semantic value of τ , if $\llbracket \tau \rrbracket_{\mathcal{A}'} = i$, then $\llbracket \phi[\tau/\xi] \rrbracket_{\mathcal{A}'} = T$.

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 $\forall xPx$ 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,

- $\llbracket \forall v\phi \rrbracket_{\mathcal{A}} = T$ iff for any constant τ , $\llbracket \phi[\tau/v] \rrbracket_{\mathcal{A}} = T$.

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 \mathcal{L}_2 , there are entities which cannot be the semantic value of any constant. Suppose we enumerate the constants a_i , letting $I_{\mathcal{A}}(a_i) = i$. 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 \mathcal{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, $\models \phi$ iff ϕ is true in every structure under every variable assignment. In effect, therefore, an open formula ϕ with free variables v_1, \dots, v_n will be a tautology iff its *universal closure* (the result of binding all free variables with universal quantifiers) $\forall v_1 \dots \forall v_n \phi$ is a tautology.

Yet this is problematic; for then this will be a correct sequent:

$$Px \models \forall xPx.$$

But the deduction theorem then fails! For

$$\models Px \rightarrow \forall x Px \quad \text{iff} \quad \models \forall x (Px \rightarrow \forall x Px),$$

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 interpretation of the quantifiers.

Exercises

1. Suppose we define a sentence ϕ to be *true*^{*} in \mathcal{A} iff $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$ for at least one variable assignment α over \mathcal{A} . Show that ϕ is true in \mathcal{A} iff it is true^{*} in \mathcal{A} .
2. Prove that, for all structures \mathcal{A} and variable assignments α , $\llbracket \exists v \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \neg \forall v \neg \phi \rrbracket_{\mathcal{A}}^{\alpha}$.
3. Prove that if v is not free in ϕ , $\llbracket \forall v \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha}$. If ϕ is a sentence, show that ϕ and $\forall v \phi$ are logically equivalent.
4. Prove that, if v and ν are variables, ϕ a formula not containing v , and $\phi[v/\nu]$ the formula that results from replacing *every* occurrence of ν in ϕ with v – *even those that occur in quantifiers* $\forall \nu$, $\exists \nu$ – then for any structure \mathcal{A} and any variable assignment α , there exists a variable assignment β on \mathcal{A} such that $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \phi[v/\nu] \rrbracket_{\mathcal{A}}^{\beta}$.
5. Prove that, for any structure \mathcal{A} , if Φ and Ψ are n -ary predicate letters such that $\llbracket \Phi \rrbracket_{\mathcal{A}} = \llbracket \Psi \rrbracket_{\mathcal{A}}$, 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 \mathcal{A} , $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \phi[\Psi/\Phi] \rrbracket_{\mathcal{A}}^{\alpha}$. Then prove that, if ϕ is a sentence, ϕ is true in \mathcal{A} iff $\phi[\Psi/\Phi]$ is true in \mathcal{A} .

6. Suppose ϕ is a formula in which only v occurs free, and τ any constant. Prove that $\forall v\phi \models \phi[\tau/v]$.
7. Suppose ϕ is a formula in which at most v occurs free, and τ any constant *not occurring in Γ or ϕ* . Prove that if $\Gamma, \phi[\tau/v] \models \psi$ then $\Gamma, \exists v\phi \models \psi$, provided τ does not occur in ψ .
8. Show that, if τ and θ are any constants:
 - (a) If $\Gamma \models \phi$ then $\Gamma[\tau/\theta] \models \phi[\tau/\theta]$.
 - (b) If $\Gamma \models$ then $\Gamma[\tau/\theta] \models$.
9. Suppose ϕ and ψ are any formulae, in which possibly (but not necessarily) v_1, \dots, v_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 $\forall v_1 \dots \forall v_n(\phi \leftrightarrow \psi) \models \delta \leftrightarrow \delta[\psi/\phi]$. (Hint: you might want to consider an induction showing joint satisfiability of any formulae γ and $\gamma[\psi/\phi]$.)
10. Show that $\forall \xi(\phi \leftrightarrow \psi) \models \forall \xi\phi \leftrightarrow \forall \xi\psi$.
11. Give a full specification of the alternative formation rules for a language \mathcal{L}_2' in which there are no open formulae. Show that the sentences of \mathcal{L}_2 are the formulae of \mathcal{L}_2' .
12. Give a semantic clause for \exists on the alternative semantics using *recursion on truth* as specified on page 147.
13. Show the following:
 - (a) $\forall x(Px \wedge Qx) \equiv \forall xPx \wedge \forall xQx$;
 - (b) $\exists x(Px \wedge Qx) \models \exists xPx \wedge \exists xQx$. Why not vice versa?
 - (c) $\exists y\forall xPxy \models \forall x\exists yPxy$. Why not vice versa?
14. If $v_1 \dots v_n$ is a sequence of variables, let $\forall v_1 \dots v_n$ be the string of quantifiers $\forall v_1 \dots \forall v_n$. Let \mathcal{P} be an arbitrary permutation of the order of a sequence. Show that $\forall v_1 \dots v_n\phi \equiv \forall \mathcal{P}v_1 \dots v_n\phi$.
15. Explain why the ‘objectual’ reading of the quantifiers (i.e., the standard one) has no difficulty with universal quantification over an uncountable domain.

Chapter 8

Natural Deduction Proofs in \mathcal{L}_2 ; Soundness

8.1 Proofs in \mathcal{L}_2

Proofs in \mathcal{L}_2 In the case of \mathcal{L}_1 , a proof procedure was not strictly speaking necessary. Every sentence of \mathcal{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 \mathcal{L}_1 -structure.

But in \mathcal{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 \mathcal{L}_2 claims from others, is crucial. Once more, we use natural deduction; again, we abuse notation at let \vdash stand for *provability in \mathcal{L}_2* .

$\frac{\forall v\phi}{\phi[\tau/v]} \forall\text{Elim}$	$\frac{\begin{array}{c} \vdots \\ \phi[\tau/v] \end{array}}{\forall v\phi} \forall\text{Intro}$	Provided τ doesn't occur in ϕ or in any undischarged assumption in the proof of $\phi[\tau/v]$.
$\frac{\phi[\tau/v]}{\exists v\phi} \exists\text{Intro}$	$\frac{\begin{array}{c} [\phi[\tau/v]] \\ \vdots \\ \psi \end{array} \quad \begin{array}{c} \vdots \\ \exists v\phi \end{array}}{\psi} \exists\text{Elim}$	Provided τ doesn't occur in $\exists v\phi$, in ψ , or in any undischarged assumption other than $\phi[\tau/v]$ in the proof of ψ .

 Table 8.1: New Natural Deduction Rules for \mathcal{L}_2

Natural Deduction The rules for the natural deduction system ND_2 include all of those from the original system ND . As \mathcal{L}_2 includes all of \mathcal{L}_1 , everything provable in ND is also provable in ND_2 . But we include new rules to cover the quantifiers, laid out in Table 8.1. Throughout, let $\phi[\tau/v]$ be the result of replacing *free* occurrences of variable v 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 78 (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 $\exists\text{Elim}$ or $\forall\text{Intro}$ (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. \square

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: $\forall x(P \vee Qx) \vdash P \vee \forall xQx$. (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 134) where the natural deduction rules for \rightarrow did not suffice to prove every tautology involving \rightarrow as its only connective.

Proof of $\forall x(P \vee Qx) \vdash P \vee \forall xQx$

$$\begin{array}{c}
 \frac{[P]}{\vdash} \text{vIntro} \\
 \frac{[\neg(P \vee \forall x Qx)] \quad P \vee \forall x Qx}{\vdash} \neg\text{Elim} \\
 \frac{\forall x(P \vee Qx)}{\vdash} \forall\text{Elim} \quad \frac{[P] \quad \neg P}{\vdash} \neg\text{Elim} \\
 \frac{P \vee Qa \quad Qa \quad [Qa]}{\vdash} \forall\text{Elim} \\
 \frac{Qa}{\vdash} \forall\text{Intro} \\
 \frac{\forall x Qx}{\vdash} \forall\text{Intro} \\
 \frac{P \vee \forall x Qx \quad [\neg(P \vee \forall x Qx)]}{\vdash} \neg\text{Elim} \\
 P \vee \forall x Qx
 \end{array}$$

Some Proof Theory: Uniqueness of \exists 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 least two prime numbers less than 8’ and ‘there are at least two cities smaller than London’, while being committed to cities, and avoiding commitment 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 79 (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 $\exists_1 v\phi$ is logically equivalent to $\exists_2 v\phi$.

Proof. Assuming that we have introduction and elimination rules for the two quantifiers, we can give the following natural deduction proof:

$$\frac{\frac{\exists_1 v\phi \quad \frac{[\phi[\tau/v]]}{\exists_2 v\phi} \exists_2 \text{Intro}}{\exists_2 v\phi} \exists_1 \text{Elim}}{\exists_2 v\phi}$$

Obviously a precisely similar proof will show that $\exists_2 v\phi \vdash \exists_1 v\phi$. \square

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.

8.2 Soundness of \mathcal{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 \mathcal{L}_2 we introduced in the last chapter. That is, we wish to prove

Theorem 80 (Soundness)

If $\Gamma \vdash \phi$ then $\Gamma \models \phi$.

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 \mathcal{L}_2 : $\forall\text{Elim}$ and $\exists\text{Intro}$ Suppose $\Gamma \vdash \phi$, where ϕ is obtained from an earlier proof by the use of the $\forall\text{Elim}$ rule, and so is of the form $\psi[\tau/v]$. Then there was a proof on the assumption Γ of $\forall v\psi$.

By the induction hypothesis, $\Gamma \models \forall v\psi$. So every structure which makes all the members of Γ true, makes $\forall v\psi$ true. By theorem 75 (proved on page 146), $\forall v\psi \models \psi[\tau/v]$. Since $\psi[\tau/v] = \phi$, ϕ will be true in every structure which makes Γ true, i.e., $\Gamma \models \phi$.

The proof for $\exists\text{Intro}$ is similar. I leave it for an exercise.

Soundness of \mathcal{L}_2 : $\forall\text{Intro}$ and $\exists\text{Elim}$ The trickier case is if $\Gamma \vdash \phi$ which was obtained from $\exists\text{Elim}$. 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, $\Gamma \models \exists v\psi$.

Since $\Theta \subseteq \Gamma$, we have $\Gamma, \psi[\tau/v] \vdash \phi$, and by the induction hypothesis $\Gamma, \psi[\tau/v] \models \phi$. By theorem 76 (proved on page 146), $\Gamma, \exists v\psi \models \phi$. And by the Cut theorem, $\Gamma \models \phi$.

The soundness of $\forall\text{Intro}$ is similarly demonstrated; I leave it for an exercise.

8.3 Completeness of \mathcal{L}_2

Completeness

Theorem 81 (Completeness of \mathcal{L}_2)

If $\Gamma \models \phi$, $\Gamma \vdash \phi$.

The proof extends the proof of completeness for \mathcal{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 \mathcal{L}_2 -sentences is satisfiable.
- Finally, show that if $\Gamma \not\models \phi$, 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 $\Gamma \not\models \phi$. By contraposition, this shows completeness.

We won't prove this theorem here; the details are involved.

Compactness**Theorem 82 (Compactness)**

If Γ is any set of \mathcal{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 \mathcal{L}_1 in Chapter 3, is more difficult in \mathcal{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 \mathcal{L}_2 must therefore fail. This is because the relation between predecessor and less than is not definable in \mathcal{L}_2 .

8.4 Decidability and Undecidability

Completeness and Decidability

Theorem 83 (\mathcal{L}_2 is positively decidable)

If $\models \phi$, 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 \mathcal{L}_2 is countable, and since every proof is of finite length, consider the proofs of length n which involve only the first m \mathcal{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 . \square

Undecidability

Theorem 84

\mathcal{L}_2 is negatively undecidable: if $\phi \not\models$, 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: $\forall x \exists y Pxy \models Paa$. 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 \mathcal{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 \mathcal{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 \mathcal{L}_2 , the halting problem would be solvable. Since it is insoluble, we have

Theorem 85 (Undecidability of \mathcal{L}_2)

\mathcal{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 \mathcal{L}_2 that are decidable?

Prenex Normal Form A sentence ϕ of \mathcal{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 86 (PNF)

Every sentence of \mathcal{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 72)
2. $\neg\exists v\phi \equiv \forall v\neg\phi$; (basically shown by Theorem 72)
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 $\forall\exists$ -sentences An $\forall\exists$ -sentence is a sentence in PNF in which all universal quantifiers precede all existential quantifiers.

Theorem 87 (Decidability of $\forall\exists$ -sentences)

There exists an effective procedure which establishes the validity of any valid $\forall\exists$ -sentence, and the invalidity of any invalid $\forall\exists$ -sentence.

The proof is as follows. We show how to reduce the question of validity of a $\forall\exists$ -sentence to the question of validity of a related quantifier-free sentence; and such quantifier-free sentences of \mathcal{L}_2 are decidable. For convenience, I abbreviate $\forall x_1 \dots \forall x_n$ as $\forall x_1 \dots x_n$, and similarly for \exists .

Lemma on Quantifier-Free Sentences

Lemma 88 (Quantifier-Free Sentences)

If ϕ is a sentence of \mathcal{L}_2 , which is not a (substitution instance of a) truth-functional tautology, then there is a structure \mathcal{A} in which $\llbracket \phi \rrbracket_{\mathcal{A}} = F$ and either (i) if ϕ contains no constants, the domain of \mathcal{A} has one element; or (ii) otherwise, \mathcal{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 \mathcal{L}_1 -structure; and we can construct an \mathcal{L}_2 -structure \mathcal{A} which agrees with that pseudo- \mathcal{L}_1 -structure on its assignment to atomic sentences. \square

A corollary is that quantifier-free sentences of \mathcal{L}_2 are decidable, since the truth-table test decides truth-functional tautologies.

\exists -sentences

Theorem 89 (Constant-free \exists -sentences)

If ϕ is a quantifier-free formula which contains only the variables v_1, \dots, v_n and which contains no constants, then $\models \exists v_1 \dots v_n \phi$ iff $\models \phi[\tau/v_1, \dots, \tau/v_n]$.

Proof. *R to L:* Repeated applications of Existential Introduction (Theorem 75) yield $\phi[\tau/v_1, \dots, \tau/v_n] \models \exists v_1 \dots v_n \phi$. By Cut, the result follows.

L to R: If $\not\models \phi[\tau/v_1, \dots, \tau/v_n]$, then by Lemma 88, there is a structure with a one-element domain \mathcal{A} where it is false. If $\llbracket \exists v_1, \dots, v_n \phi \rrbracket_{\mathcal{A}} = T$, there is a variable assignment α such that $\llbracket \phi \rrbracket_{\mathcal{A}}^{\alpha} = T$; but this variable assignment

must assign $\llbracket \tau \rrbracket_{\mathcal{A}}^\alpha$ to each variable (there is only one element in the domain), so $\llbracket \phi[\tau/v_1, \dots, \tau/v_n] \rrbracket_{\mathcal{A}}^\alpha = T$, contradiction. (Using Theorem 74.) \square

These \exists -sentences are *equi-decidable* with quantifier-free sentences.

Theorem 90 (Arbitrary \exists -sentences)

If ϕ is a quantifier free formula containing only τ_1, \dots, τ_n and v_1, \dots, v_m , then

$$\models \exists v_1 \dots v_m \phi \quad \text{iff} \quad \models \bigvee (\phi[\tau_i/v_j]),$$

where $\bigvee (\phi[\tau_i/v_j])$ is the disjunction of all the ways of substituting the constants τ_1, \dots, τ_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 \exists -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 $\forall\exists$ -sentence is valid iff some \exists -sentence is valid.

$\forall\exists$ -Sentences

Theorem 91 ($\forall\exists$ -sentences and \exists -sentences)

If τ_1, \dots, τ_m don't occur in ϕ , then

$$\models \forall v_1 \dots v_m \exists v_1 \dots v_n \phi \quad \text{iff} \quad \models \exists v_1 \dots v_n \phi[\tau_1/v_1, \dots, \tau_m/v_m].$$

Proof. L to R: By repeated applications of \forall -Elimination (Theorem 75), we have

$$\forall v_1 \dots v_m \exists v_1 \dots v_n \phi \models \exists v_1 \dots v_n \phi[\tau_1/v_1, \dots, \tau_m/v_m],$$

from which the result follows by Cut.

R to L: Because Γ is empty, and τ_1, \dots, τ_m don't occur in ϕ , we can use \forall -Introduction (Theorem 76) repeatedly to show the theorem. \square

So any $\forall\exists$ -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., $\forall x\exists yRxy$ can't be put into that form.

An \mathcal{L}_2 sentence is *monadic* iff it contains at most unary (one-place) predicates.

Lemma 92 (Monadic and Elementary Sentences)

Every monadic sentence of \mathcal{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. \square

Corollary: Monadic \mathcal{L}_2 decidable

Theorem 93 (Monadic \mathcal{L}_2 sentences)

If ϕ is a monadic sentence of \mathcal{L}_2 , it is equivalent to a $\forall\exists$ -sentence.

Proof. Suppose ϕ is a monadic sentence. We can effectively produce 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 \mathcal{L}_2 is decidable. \square

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 \mathcal{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 \forall Intro:

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \phi \end{array}}{\forall v \phi[v/\tau]} \forall\text{Intro} \quad \text{Provided } \tau \text{ doesn't occur in any undischarged assumption in the proof of } \phi.$$

3. Let ψ be the result of substituting v for *all* occurrences of τ in ϕ . Show that this rule is not sound (i.e., not truth-preserving)

If $\Gamma, \phi \vdash \chi$ then $\Gamma, \exists v \psi \vdash \chi$, provided τ does not occur in Γ or in χ .

Can you think of a way of amending the rule to make it sound? (*Hint: Theorem 7.3.*)

4. Verify that all the rules of *ND* are sound with respect to the \mathcal{L}_2 semantics.
5. (a) Show that the rule \exists Intro is sound with respect to the \mathcal{L}_2 semantics.
(b) Show that the rule \forall Intro is sound with respect to the \mathcal{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 \mathcal{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 \mathcal{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 \mathcal{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 \mathcal{L}_2 ?
10. Prove the Lemma on Quantifier-Free Sentences.
11. Prove the Theorem for Arbitrary \exists -Sentences.
12. Suppose that ϕ is a monadic \mathcal{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 $P \leftrightarrow \forall x Qx$. 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 9

Syntax, Semantics, and Proofs in $\mathcal{L}_=$

9.1 Identity

Identity: Syntax and Semantics The language \mathcal{L}_2 doesn't have a privileged predicate for identity. Yet such a predicate is very useful in discussing binary relations.

We *add* to \mathcal{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 \mathcal{L}_2 , but give this new predicate a constant semantic value, so that in every \mathcal{L}_2 -structure \mathcal{A} ,

$$(=) \quad I_{\mathcal{A}}(=) = \{\langle x, x \rangle : x \in D_{\mathcal{A}}\}$$

Of course the property that we intend this predicate to express could exist in an \mathcal{L}_2 -structure already; the change is that we now give it a logically privileged meaning. The satisfaction conditions are:

- $\llbracket \tau_1 = \tau_2 \rrbracket_{\mathcal{A}}^{\alpha} = T$ iff $\langle \llbracket \tau_1 \rrbracket_{\mathcal{A}}^{\alpha}, \llbracket \tau_2 \rrbracket_{\mathcal{A}}^{\alpha} \rangle \in \llbracket = \rrbracket_{\mathcal{A}}^{\alpha}$, i.e., iff $\llbracket \tau_1 \rrbracket_{\mathcal{A}}^{\alpha} = \llbracket \tau_2 \rrbracket_{\mathcal{A}}^{\alpha}$; i.e., if $\llbracket \tau \rrbracket_{\mathcal{A}}^{\alpha}$ is the very same object as $\llbracket \kappa \rrbracket_{\mathcal{A}}^{\alpha}$.

Call the language which keeps all the old atomic formulae and formation rules of \mathcal{L}_2 , but adds these atomic formulae, $\mathcal{L}_=$.

Some Theorems About Identity

Theorem 94 (Theoremhood of Identity)

For any constant τ , $\models \tau = \tau$.

Proof. Suppose $\not\models \tau = \tau$ for some τ . Then there is a structure \mathcal{A} such that $\llbracket \tau = \tau \rrbracket_{\mathcal{A}} = F$, iff $\llbracket \tau \rrbracket_{\mathcal{A}}^\alpha \neq \llbracket \tau \rrbracket_{\mathcal{A}}^\alpha$; impossible. \square

Theorem 95 (Substitution of co-designating terms II)

$\tau = \kappa \models \phi[\tau/v] \leftrightarrow \phi[\kappa/v]$.

Proof. Fairly immediate from the Substitution of Co-Designating Terms theorem (chapter 5, page 144).

The Relation of Identity

Theorem 96 (Identity an Equivalence Relation)

- $\models \forall x x = x$;
- $\models \forall x \forall y (x = y \rightarrow y = x)$;
- $\models \forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$.

Theorem 97 ('Leibniz' Law')

For any ϕ in which at most v occurs free, $\models \forall v \forall v (v = v \rightarrow \phi \leftrightarrow \phi[v/v])$.

Proofs left for exercises. *Leibniz' Law* usually refers to this definition of identity in second order logic: $\forall v \forall v (v = v \leftrightarrow \forall \Phi (\Phi(v) \leftrightarrow \Phi[v/v](v)))$.

Proofs in $\mathcal{L}_=$ To obtain a natural deduction system for $\mathcal{L}_=$, $ND_=$, we add to the rules of ND_2 the rules in Table 9.1, keeping all other rules. In =Elim-l and =Elim-r, ϕ is a formula in which at most v occurs free.

$$\begin{array}{c}
 \frac{[\tau = \tau]}{} =\text{Intro} \\
 \vdots \\
 \frac{\phi[\tau/v] \quad \tau = \kappa}{\phi[\kappa/v]} =\text{Elim-r} \quad \frac{\phi[\tau/v] \quad \kappa = \tau}{\phi[\kappa/v]} =\text{Elim-l}
 \end{array}$$

 Table 9.1: New Natural Deduction Rules for $\mathcal{L}_=$

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 dispensable.

Theorem 98 (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.

$$\begin{array}{c}
 \frac{[\kappa = \kappa] \quad \tau = \kappa}{\kappa = \tau} =\text{Elim-l} \\
 \text{Proof. } \frac{\phi[\tau/v] \quad \kappa = \tau}{\phi[\kappa/v]} =\text{Elim-l}
 \end{array}$$

This is obviously equivalent to =Elim-r; a similar proof would show the dispensibility of =Elim-l. (Note the use of =Intro.) \square

Soundness of $\mathcal{L}_=$

Theorem 99 ($\mathcal{L}_=$ is Sound)

If ϕ is a sentence of $\mathcal{L}_=$, then if $\vdash_{\mathcal{L}_=} \phi$ then $\models \phi$.

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, $\llbracket \tau = \tau \rrbracket_{\mathcal{A}} = T$ in every structure, and by the induction hypothesis $\llbracket \phi \rrbracket_{\mathcal{A}} = T$ if $\llbracket \tau = \tau \rrbracket_{\mathcal{A}} = T$, clearly $\llbracket \phi \rrbracket_{\mathcal{A}} = T$.

- We have a proof of $\tau = \kappa$ on assumptions Γ , and a proof of $\phi[\tau/v]$ on assumptions Δ , and we extend to a proof of $\phi[\kappa/v]$ using =Elim-r on assumptions $\Gamma \cup \Delta$. By the induction hypothesis, for any \mathcal{A} where $\Gamma \cup \Delta$ is satisfied, $\llbracket \tau = \kappa \rrbracket_{\mathcal{A}} = T$, and similarly $\llbracket \phi[\tau/v] \rrbracket_{\mathcal{A}} = T$. By the Substitution of co-designating constants theorem of page 167 and logic, in any such \mathcal{A} , $\llbracket \phi[\kappa/v] \rrbracket_{\mathcal{A}} = T$. \square

Further metalogical results It turns out that the proof of completeness of \mathcal{L}_2 can be extended to provide a completeness proof for $\mathcal{L}_=$.

Theorem 100 (Completeness of $\mathcal{L}_=$)

If $\models \phi$ then $\vdash \phi$.

In the now familiar way, this can be adapted to a compactness proof.

Since \mathcal{L}_2 is not decidable, and $\mathcal{L}_=$ contains \mathcal{L}_2 as a part, $\mathcal{L}_=$ is not decidable either. Obviously the monadic fragment of $\mathcal{L}_=$ is decidable (as ‘=’ is binary, the monadic fragment is the same as that of \mathcal{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).

9.2 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 P s, 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 ϕ s’ 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 $\mathcal{L}_=$ 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 $\mathcal{L}_2/\mathcal{L}_=$ 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 94 (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 $\mathcal{L}_=$, 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', $<$ be 'is earlier than', P be 'is prime minister at', and A be 'is Australian'. (*) has two readings:

$$\begin{aligned} \forall x(x < t \rightarrow \exists y((Pyx \wedge \forall z(Pzx \leftrightarrow y = z))) \wedge Ay); \\ \exists y((Pyt \wedge \forall z(Pzt \leftrightarrow y = z)) \wedge \forall x(x < t \rightarrow Ay)). \end{aligned}$$

The first says: always the prime minister, 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 (\ddagger) 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 (\ddagger) 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 apparently 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 \mathcal{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.

9.3 Compactness and Cardinality

‘There are n things’ and infinity We saw before that there is a set of $\mathcal{L}_=$ sentences satisfiable only in infinite domains. If R is a transitive asymmetric relation (both of which properties are expressible by an $\mathcal{L}_=$ 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 $\mathcal{L}_=$ sentence which is an axiom of infinity (exercise).

Finitude Undefinable Since there is at least one sentence of $\mathcal{L}_=$ true only in infinite domains, perhaps there is a sentence of $\mathcal{L}_=$ true only in *finite* domains? Perhaps surprisingly, there is not.

Theorem 101 (Finitude Undefinable)

There is no set of $\mathcal{L}_=$ -sentences satisfiable in all and only finite domains.

Proof. Suppose FIN is a set of $\mathcal{L}_=$ 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 \mathcal{A} with a domain of size greater than n , Δ is satisfiable in \mathcal{A} . Since $\Gamma \cup \Delta$ is unsatisfiable, Γ is unsatisfiable in any such model \mathcal{A} , so there is a model with a finite domain (any such \mathcal{A}) in which FIN is unsatisfiable, contrary to assumption. So there is no such FIN .

Löwenheim-Skolem Theorem**Theorem 102 (The Löwenheim-Skolem Theorem)**

If Γ is a set of sentences of $\mathcal{L}_=$ which has an infinite model \mathcal{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 \mathcal{L}^+ by adding new constants C such that size of C is ω_β . Consider the set of \mathcal{L}^+ sentences $\Gamma^* = \Gamma \cup \{c_i \neq c_j : i \neq j, c_i, c_j \in C\}$. 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). \square

Overspill**Theorem 103 (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 $n \in \Delta$. 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). \square

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 theorem says it has a

(presumably 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)

9.4 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 $\mathcal{L}_=$ or \mathcal{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 $\neg\neg\phi \not\equiv \phi$; 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 \times . 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 104 (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 \mathcal{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 and the topic of non-classical 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 167.
2. Prove the following
 - (a) $\models \forall x x = x$;
 - (b) $\models \forall x \forall y (x = y \rightarrow y = x)$;
 - (c) $\models \forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$.
3. (a) Show that in the presence of $=\text{Elim}$, $\exists\text{Elim}$ and $\exists\text{Intro}$, the rule $=\text{Intro}$ can be replaced by this rule (where τ is a constant):

$$\frac{[\exists v v = \tau]}{\vdots} =\text{Intro}\dagger$$

- (b) What would be the effect of adopting, instead of our rule $=\text{Intro}$, this rule (in the presence of the other rules of ND_2):

$$\frac{\phi}{\tau = \tau} =\text{Intro}^*$$

- (c) Can we replace both $=\text{Intro}$ and $=\text{Elim}$ by this pair of rules, in the presence of the other rules of ND_2 (ϕ is a formula in which at most v occurs free):

$$\frac{\phi[\tau/v]}{\exists v(v = \tau \wedge \phi)} =\text{Intro}\ddagger \quad \frac{\exists v(v = \tau \wedge \phi)}{\phi[\tau/v]} =\text{Elim}\ddagger$$

4. Suppose ϕ is a formula in which at most v occurs free, and in which v does not occur at all.
 - (a) Show that $\models \forall v \forall v (v = v \rightarrow \phi \leftrightarrow \phi[v/v])$.
 - (b) Suppose for every formula ϕ with just v free, not itself involving identity, and any constants τ, κ , we could show that $\llbracket \phi[\tau/v] \rrbracket_{\mathcal{A}} = \llbracket \phi[\kappa/v] \rrbracket_{\mathcal{A}}$. Could we then show that $\llbracket \tau = \kappa \rrbracket_{\mathcal{A}} = T$? (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 $\exists_{n+1}xPx \equiv \exists x(Px \wedge \exists_n y(Py \wedge \neg x = y))$.
 (b) Define $\forall_n x\phi$ as $\neg\exists_n x\neg\phi$. Give an English translation of $\forall_n xPx$.
6. In the analysis of definite descriptions, $\exists v(\psi \wedge (\forall v\phi[v/v] \leftrightarrow v = v))$, could we replace the ' \leftrightarrow ' by ' \rightarrow '? If not, why? What about in this formulation: $\exists v(\psi \wedge (\forall v v = v \leftrightarrow \phi[v/v]))$?
7. Give appropriate analyses in $\mathcal{L}_=$ 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 $\mathcal{L}_=$? What are the difficulties with a Russellian analysis of (\P) '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 $\mathcal{L}_=$ -sentence which is satisfied in \mathcal{A} iff \mathbf{N} is satisfied in \mathcal{A} .
12. A *permutation* on S is a one-one function π from S to S . If \mathcal{A} is a model, where π is a function from the domain of \mathcal{A} to itself, let the permutation $\pi(\mathcal{A})$ be this model
 - $D_{\mathcal{A}} = D_{\pi(\mathcal{A})}$.
 - For any term τ , $I_{\pi(\mathcal{A})}(\tau) = \pi(I_{\mathcal{A}}(\tau))$.

- For any n -ary predicate Φ ,

$$I_{\pi(\mathcal{A})}(\Phi) = \{ \langle \pi(x_1), \dots, \pi(x_n) \rangle : \langle x_1, \dots, x_n \rangle \in I_{\mathcal{A}}(\Phi) \}.$$

- (a) Show the *Permutation Theorem*: for any sentence ϕ , and permutation π , $\llbracket \phi \rrbracket_{\pi(\mathcal{A})} = \llbracket \phi \rrbracket_{\pi(\mathcal{A})}$.
- (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?

Chapter 10

Logic and Natural Language: Conditionals, Entailment, Designators, and Relations

10.1 Conditionals

‘If ϕ , ψ ’ and \rightarrow So far we have rendered the English conditional construction ‘If ϕ , ψ ’ 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. Consider:

- (4) The Tigers will not lose every game this season;
- (5) The Tigers will make it to the finals;
- (6) Therefore: If the Tigers lose every game this season,
they will make it to the finals.

If ‘if ϕ , ψ ’ is $\phi \rightarrow \psi$, this argument should be valid (it is an instance of the valid form $\phi, \psi \vdash \neg\phi \rightarrow \psi$). But this argument does not strike us as valid.

Can we do better in our logic? The first question we should address is: is there a better option in \mathcal{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*: there exists at least one structure \mathcal{A} where $f(|\phi|_{\mathcal{A}}, |\psi|_{\mathcal{A}}) \neq f(|\psi|_{\mathcal{A}}, |\phi|_{\mathcal{A}})$. 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 boxed argument above that it is undesirable for the truth of a conditional to follow from the truth of the consequent or 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: \rightarrow and a function g (expressed by $\neg(\psi \rightarrow \phi)$). 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 \rightarrow .

Meaning as characterised by derivation rules Of course, this is not a strong argument – that \rightarrow is better than other candidates is no argument that \rightarrow is itself adequate! But we can offer better considerations. Given that our derivation system for \mathcal{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 \mathcal{L}_1 , notably the rules governing ‘and’. (This is some motivation behind the nomenclature of ‘natural deduc-

tion' – 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 conditional connective '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 ψ can be established, conditional on the assumption that ϕ , then you can establish 'if ϕ , ψ ' without need of that assumption.

It is obvious that something which didn't obey these rules would be quite unlike 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, as seems very plausible, these two rules capture the English conditional, 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 $\phi \rightarrow \psi$. This is equivalent to $\neg\phi \vee \psi$. Assume ϕ . By disjunctive syllogism, ψ . By conditional proof, it follows that if ϕ , ψ , without need of the assumption.

'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 \mathcal{L}_1 ; it is obviously a good rule for the English 'or'.)

Replying to the apparent counterexamples – Grice If this argument works, we're wrong to think that the cases where \rightarrow doesn't seem to behave like 'If...then' are genuine counterexamples. But we need to say something about those cases.

The explanation offered is that, even though these sentences are true, they *con conversationally implicate* something false, and that is what we are responding to. The notion of conversational implicature was introduced by Grice, who noted that many utterance 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.

Applying Quantity to conditionals 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.

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

Theorem 105 (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 $\phi \models \psi$ 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 support for it can be found by considering cases in which the application of Import-Export seems unproblematic. Consider this pair, which seem to express the same claim:

- (7) a. 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.
- b. If you drink one more can of beer and I drink one more can of beer then we’ll be completely out of beer. (Kratzer, 2012: 88)

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:

- (8) If a Republican wins and it's not Reagan, then it's Anderson who wins.

(8) is a conditional with a conjunctive antecedent; if Import-Export is valid for the English 'if', then we can derive

- (9) 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 (9) - 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:

- (10) 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 ((8) and that a Republican will win), using Import-Export and *modus ponens*, a claim we believe to be false: 10. 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 (8), 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.

10.2 Entailment

Another problematic feature of \mathcal{L}_1 (somewhat related to the foregoing by the deduction theorem) are the so-called *paradoxes of entailment*. These are the following valid sequents:

- $\phi, \neg\phi \models \psi$;
- $\phi \models \psi \vee \neg\psi$.

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.

(11)	ϕ	Assumption
(12)	ϕ or ψ	Disjunction introduction, 1
(13)	$\neg\phi$	Assumption
(14)	ψ	Disjunctive syllogism, 2, 3

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 syl-

logism, 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’: $\phi, \psi \models \phi \wedge \psi$ is correct, ‘possibly ϕ , possibly $\psi \models$ possibly $\phi \wedge \psi$ ’ is not correct – think of the case where ψ = ‘the coin toss lands heads’, ϕ = ‘the coin toss lands tails’.)

10.3 Designators

Designators in \mathcal{L}_2 and English The formal properties of \mathcal{L}_2 have been established; now we begin a relatively informal investigation into the possible connections between the semantics of \mathcal{L}_2 and the semantics of English. We begin in this section with a discussion of *designation* in both languages.

Direct Reference \mathcal{L}_2 embodies a particularly simple view of the meaning of a constant.

Definition 95 (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 \mathcal{L}_2 directly refer, because the meaning (semantic value) of a in \mathcal{A} is $\llbracket a \rrbracket_{\mathcal{A}}$, which is just an item in the domain of \mathcal{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 9) 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 96 (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 \mathcal{L}_2 , because if $\mathcal{A} \neq \mathcal{B}$, then it may be that $I_{\mathcal{A}}(a) \neq I_{\mathcal{B}}(a)$. 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 \mathcal{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

¹Be 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.

– which is just some object **a**, given direct reference – satisfies F at each possibility in which **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.

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 \mathcal{L}_2 .

On the one hand, there is considerable syntactic and semantic evidence that names in English function rather like the constants of \mathcal{L}_2 . Consider

(15) Antony Eagle is lecturing now.

(15) 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 \mathcal{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 Hes-

perus 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?:

(16) Hesperus is Hesperus.

(17) 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 $\mathcal{L}_=$ 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 $\mathcal{L}_=$, 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

(18) Lois Lane believes that Superman is Superman;

(19) 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 \mathcal{L}_2 lacks opaque contexts, as theorems about substitution of co-referring constants proved in Chapter 7 show. This is an expressive weakness of \mathcal{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 \mathcal{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 \mathcal{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.

10.4 More on Relations

Expressing Properties of Binary Relations in $\mathcal{L}_{2=}$ We talked about the properties of binary relations, modelled as sets, back in chapter 2.

We can use formulae of \mathcal{L}_2 to express various properties of relations on an entire domain. For example

Reflexivity A relation R is reflexive on $D_{\mathcal{A}}$ iff $R = \llbracket P^2 \rrbracket_{\mathcal{A}}$ and $\llbracket \forall x P^2_{xx} \rrbracket_{\mathcal{A}} = T$.

Transitivity A relation R is transitive on $D_{\mathcal{A}}$ iff $R = \llbracket P^2 \rrbracket_{\mathcal{A}}$ and

$$\llbracket \forall x \forall y \forall z (P^2_{xy} \wedge P^2_{yz} \rightarrow P^2_{(xz)}) \rrbracket_{\mathcal{A}} = T.$$

Symmetry A relation R is symmetric on $D_{\mathcal{A}}$ iff $R = \llbracket P^2 \rrbracket_{\mathcal{A}}$ and $\llbracket \forall x \forall y (P^2_{xy} \rightarrow P^2_{yz}) \rrbracket_{\mathcal{A}} = T$.

Relations in English and \mathcal{L}_2 : Intension and Extension Is the semantic value of an English predicate expression a property or relation as those are interpreted in \mathcal{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 \mathcal{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 \mathcal{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 \mathcal{L}_2 is rather weaker than that against English having a similar semantics for designators.

Further Reading

Beall and van Fraassen (2003) and Burgess (2009), cited in the last chapter, 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.

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 $\models \phi$ and $\models \phi \rightarrow \psi$, then $\models \psi$.
 - ii. *Disjunctive Syllogism*: If $\models \phi \vee \psi$ and $\models \neg\phi$, then $\models \psi$.
 - iii. *Reductio ad Absurdum*: If $\phi \models \psi$ and $\phi \models \neg\psi$ then $\models \neg\phi$.
- (b) In our system each acceptable rule of inference has a corresponding correct semantic sequent (for example, the sequent $\phi, (\phi \rightarrow \psi) \models \psi$

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 ‘ $\phi \models_{\text{English}} \text{Necessarily } \phi$ ’ intuitively correct?

2. Give the proof of Gibbard’s theorem (Theorem 105). (*Hint*: begin with the fact that all instances of this schema for disjunctive syllogism are valid: $((\neg\phi \vee \psi) \wedge \phi) \models \psi$.)
3.
 - (a) John argues that, since ϕ and ψ together entail (in English) ψ , it 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?
4. Say that $\phi \models \psi$ 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 if $\phi \models \psi$ is a perfectible sequent. (*Hint*: use the Craig Interpolation Theorem (page 97), plus properties of substitution.)
 - (b) Give examples to show that perfectible entailment is not transitive (i.e., that there are cases where ϕ perfectly entails ψ , where ψ perfectly entails χ , but where ϕ does not perfectly entail χ .)
 - (c) Which of the structural rules of §4.6 (page 72) are perfectly inappropriate (i.e., take perfectible sequents to non-perfectible sequents.)?
5. Which of the relations expressed by the following English predicates are equivalence relations:
 - (a) ‘ x entails y ’, on the domain of sentences of \mathcal{L}_2 .
 - (b) ‘ x is logically equivalent to y ’, on the domain of sentences of \mathcal{L}_2 .
 - (c) ‘If $\{x\}$ is consistent, then $\{x\} \cup \{y\}$ is consistent’ on the domain of sentences of \mathcal{L}_2 .

6. Show that $\forall x \forall y (\neg x = y \rightarrow (Pxy \vee Pyx)) \equiv \forall x \forall y (x = y \vee Pxy \vee Pyx)$.
7. Show by suitable reasoning that in a finite domain, for any *partial order* R , $\exists x \forall y (Ryx \rightarrow x = y)$. Give a counterexample to this condition in an infinite domain.
8. Give a graph, on some non-empty domain, of a relation R which satisfies this condition: $\forall x \forall y (Rxy \rightarrow \exists z ((x \neq z) \wedge (y \neq z) \wedge Rxz))$. Can you give an example of a relation which satisfies this condition (being sure to specify the domain)?
9. 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* symmetric:

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 $\text{not-}P$ 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 $\text{not-}P$ is true. But then ' P or $\text{not-}P$ ' 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 $\text{not-}P$, 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 $\text{not-}P$ ' is true. Does this mean that either P is true or ' $\text{not-}P$ ' 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 42

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 $Y \not\subseteq X$.
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) $\langle x, y \rangle = \langle u, v \rangle$ 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 $\langle x, y \rangle = \{\{x\}, \{x, x\}\} = \{\{x\}, \{x\}\} = \{\{x\}\}$.
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., $\text{succ}(x) = x + 1$) is into \mathbb{N} but not onto, since there is no n such that $\text{succ}(n) = 0$ but $0 \in \mathbb{N}$.
- ii. The identity function from natural numbers into integers $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$ 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, $\text{sq}(x) = x^2$ for any $x \in \mathbb{Z}$.

Since $\text{sq}(2) = \text{sq}(-2) = 4$, 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 4. $n = \{\langle T, F \rangle, \langle F, T \rangle\}$; 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 55

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 \mathcal{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 ‘ \neg ’, 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 occurrence 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 19, 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 19, since ψ is a sentence, χ is not, contradiction.
- So $\ulcorner \oplus = \otimes \urcorner$. But that would contradict our initial 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 \mathcal{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 \mathcal{L}_1 is uniquely readable.

Exercises for Chapter 4: page 83

1. $\Gamma \models \phi$ iff every \mathcal{L}_1 -structure \mathcal{A} in which each member of Γ is true is also one in which ϕ is true; iff every \mathcal{L}_1 -structure \mathcal{A} in which each member of Γ is true is also one in which $\neg\phi$ is false; iff every \mathcal{L}_1 -structure \mathcal{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 $\Gamma \models \phi$, 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 $\llbracket \chi \rrbracket_{\mathcal{A}} = F$ iff either $\llbracket \phi \rrbracket_{\mathcal{A}} = F$ or $\llbracket \psi \rrbracket_{\mathcal{A}} = F$ iff either $\llbracket \phi \rrbracket_{\mathcal{A}} \neq T$ or $\llbracket \psi \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \phi \wedge \psi \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \chi \rrbracket_{\mathcal{A}} \neq T$.

- (b) Suppose $\chi = (\phi \rightarrow \psi)$. Then $\llbracket \chi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} = T$ and $\llbracket \psi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} \neq F$ and $\llbracket \psi \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \phi \rightarrow \psi \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \chi \rrbracket_{\mathcal{A}} \neq T$.
- (c) Suppose $\chi = (\phi \leftrightarrow \psi)$. Then $\llbracket \chi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} \neq \llbracket \psi \rrbracket_{\mathcal{A}}$ iff $\llbracket (\phi \leftrightarrow \psi) \rrbracket_{\mathcal{A}} \neq T$ iff $\llbracket \chi \rrbracket_{\mathcal{A}} \neq T$.
9. If ϕ and ψ express the same truth-function f_n , and they contain the same sentence letters s_1, \dots, s_n then in every structure \mathcal{A} ,

$$f(\mathcal{A}(s_1), \dots, \mathcal{A}(s_n)) = \llbracket \phi \rrbracket_{\mathcal{A}} = \llbracket \psi \rrbracket_{\mathcal{A}}.$$

But then these two sentence have the same truth value in every \mathcal{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 k , but are not logically equivalent.

If ϕ and ψ contain the same n sentence letters and are logically equivalent, then in every \mathcal{L}_1 -structure \mathcal{A} , $\llbracket \phi \rrbracket_{\mathcal{A}} = \llbracket \psi \rrbracket_{\mathcal{A}}$. If ϕ expresses f , then $f(\mathcal{A}(s_1), \dots, \mathcal{A}(s_n)) = \llbracket \phi \rrbracket_{\mathcal{A}} = \llbracket \psi \rrbracket_{\mathcal{A}}$. But since ψ contains just s_1, \dots, s_n too, ψ also expresses f .

10. The underlying result is this: for any structure \mathcal{A} ,

$$\llbracket \neg(\phi \leftrightarrow \psi) \rrbracket_{\mathcal{A}} = \llbracket (\neg\phi \leftrightarrow \psi) \rrbracket_{\mathcal{A}}.$$

For $\llbracket \neg(\phi \leftrightarrow \psi) \rrbracket_{\mathcal{A}} = T$ iff $\llbracket \phi \leftrightarrow \psi \rrbracket_{\mathcal{A}} = F$ iff $\llbracket \phi \rrbracket_{\mathcal{A}} \neq \llbracket \psi \rrbracket_{\mathcal{A}}$ iff $\llbracket \neg\phi \rrbracket_{\mathcal{A}} = \llbracket \psi \rrbracket_{\mathcal{A}}$ iff $\llbracket \neg\phi \leftrightarrow \psi \rrbracket_{\mathcal{A}} = T$.

Suppose we have some sentence of our language $\mathcal{L}_{\leftrightarrow, \neg}$, which has a constituent which is a negated biconditional. We can apply the above result, and Theorem 41, 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 112

1. (b) $\llbracket \phi \wedge \psi \rrbracket_{\mathcal{A}} = \max(\llbracket \phi \rrbracket_{\mathcal{A}}, \llbracket \psi \rrbracket_{\mathcal{A}}) = \max(\llbracket \psi \rrbracket_{\mathcal{A}}, \llbracket \phi \rrbracket_{\mathcal{A}}) = \llbracket \psi \wedge \phi \rrbracket_{\mathcal{A}}$, by commutativity of ‘max’.
- (c) \mathbf{k} is commutative and associative. For, e.g., $\mathbf{k}(\llbracket \phi \rrbracket_{\mathcal{A}}, \llbracket \psi \rrbracket_{\mathcal{A}}) = \llbracket \phi \wedge \psi \rrbracket_{\mathcal{A}} = \llbracket \psi \wedge \phi \rrbracket_{\mathcal{A}} = \mathbf{k}(\llbracket \psi \rrbracket_{\mathcal{A}}, \llbracket \phi \rrbracket_{\mathcal{A}})$. 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 $\{s_1, \dots, s_n\}$, and construct a sentence using them as follows.

- If in some structure \mathcal{A} , $f(\llbracket s_1 \rrbracket_{\mathcal{A}}, \dots, \llbracket s_n \rrbracket_{\mathcal{A}}) = F$, call \mathcal{A} an f -disagreeing structure. If \mathcal{A} is an f -disagreeing structure, let

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

For each f -disagreeing structure \mathcal{A} , we can see that $\llbracket S_i^{\mathcal{A}} \rrbracket_{\mathcal{A}} = F$.

- Define $\mathbf{d}_{\mathcal{A}} = \left[\bigvee_{i=1}^n S_i^{\mathcal{A}} \right]$ for each f -disagreeing structure \mathcal{A} . Since in each f -disagreeing structure, the disjuncts of this disjunction are all

F , the disjunction $\mathfrak{d}_{\mathcal{A}}$ also is F in an f -disagreeing structure \mathcal{A} . Indeed, by construction, $\mathfrak{d}_{\mathcal{A}}$ is false only in \mathcal{A} and structures that agree with \mathcal{A} on sentence letters in $\{s_1, \dots, s_n\}$ – 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 sentence letters in $\{s_1, \dots, s_n\}$; for each such class which contains an f -disagreeing structure, pick some f -disagreeing structure from it at random. There are finitely many of these chosen f -disagreeing structures which can be enumerated $\mathcal{A}_1, \dots, \mathcal{A}_m$.

- Define $\mathfrak{c} = \left[\bigvee_{j=1}^m \mathfrak{d}_{\mathcal{A}_j} \right]$. 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, \dots, s_n where f gets the value F .

5. Recall that a truth-function is positive iff $f(T, \dots, T) = T$. 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 $f(\llbracket s \rrbracket_{\mathcal{A}}) = \mathcal{A}(s)$. So if $\mathcal{A}(s) = T$, $f(T) = T$. So f is positive.

Induction step. ϕ is complex (a conjunction or conditional), and the induction hypothesis holds of its simpler constituents ψ and χ . Then ψ and χ both express positive truth functions, f_ψ and f_χ – so $f_\psi(T, \dots, T) = T = f_\chi(T, \dots, T)$.

- $\phi = (\psi \wedge \chi)$. ϕ then expresses the truth function $\mathbf{k}(f_\psi(\dots), f_\chi(\dots))$. Since those constituent truth functions are positive,

$$\mathbf{k}(f_\psi(T, \dots, T), f_\chi(T, \dots, T)) = \mathbf{k}(T, T) = T.$$

So ϕ expresses a positive truth function too.

- $\phi = (\psi \rightarrow \chi)$. ϕ then expresses the truth function $c(f_\psi(\dots), f_\chi(\dots))$. Since those constituent truth functions are positive,

$$c(f_\psi(T, \dots, T), f_\chi(T, \dots, T)) = c(T, T) = T.$$

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 constant 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) \perp is not expressively adequate by itself, since it is not true in any structure and so cannot express any positive truth-function. However, $\{\rightarrow, \perp\}$ is expressively adequate: $(P \rightarrow \perp)$ expresses the same truth-function as $\neg P$ (because that conditional expresses the function that yields F iff only if P is true and \perp is false, but since \perp is always false, iff P is true). And it can be easily verified that $((P \rightarrow \perp) \rightarrow Q)$ expresses disjunction $(P \vee Q)$. And conjunction isn't expressively adequate itself.

7. (a) We can express negation using \rightarrow and \rightarrow^* as follows: $P \rightarrow^* (P \rightarrow P)$ (or $P \rightarrow (P \rightarrow^* P)$). Since $\{\neg, \rightarrow\}$ are expressively adequate, so too is $\{\rightarrow, \rightarrow^*\}$.

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 $f^*(x^*, y^*) = z^*$. If $f = f^*$, so that f is self-dual, then if $f(T, F) = z$, $f(F, T) = z^*$, 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 $\Gamma \models^* \Delta$, 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 $\tilde{\Delta}$ be the set which results from taking every member of Δ and negating it (i.e., $\tilde{\Delta} = \{\neg\delta : \delta \in \Delta\}$). Then $\Gamma \cup \tilde{\Delta}$ is unsatisfiable.

By Compactness, some finite subset of $\Gamma \cup \tilde{\Delta}$ is unsatisfiable – call that set $G\tilde{D}$. There are three cases to consider:

- $G\tilde{D}$ contains elements from both Γ and $\tilde{\Delta}$. Let Φ be the conjunction of all sentences in both Γ and $G\tilde{D}$. Since the latter is finite, the conjunction is finite: $(\phi_1 \wedge \dots \wedge \phi_m)$. Let Ψ' be the conjunction of all sentences in $\tilde{\Delta}$ and $G\tilde{D}$. Since Ψ' is a conjunction of negated sentences $(\neg\psi_1 \wedge \dots \wedge \neg\psi_n)$, it is logically equivalent to a negated disjunction $\neg(\psi_1 \vee \dots \vee \psi_n) = \neg\Psi$. (Note that Ψ is the embedded disjunction.) Since $G\tilde{D}$ is unsatisfiable, $\Phi, \neg\Psi \models$; by Theorem 25, $\Phi \models \Psi$, and Φ and Ψ are of the right form.
 - $G\tilde{D}$ contains elements only from Γ . Let Φ be the conjunction of every sentence in $G\tilde{D}$, and let Ψ be an arbitrary finite disjunction of members from Δ . Since $G\tilde{D} \subseteq \Gamma$, Φ is of the right form, and since Φ is a contradiction, $\Phi \models \Psi$.
 - $G\tilde{D}$ contains elements only from $\tilde{\Delta}$. Then every structure makes some member of Δ true whose negation is in $G\tilde{D}$. Let Ψ be the disjunction of negations of every sentence in $G\tilde{D}$; this is a finite tautologous disjunction. Let Φ be an arbitrary finite conjunction of members from Γ . Since Ψ is a tautology, $\Phi \models \Psi$.
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 21, 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 determines 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 $f_n(n) \neq 1$, then set $d(n) = 1$.

If g is a good test for effective procedures, every function in this little flowchart 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.

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	v, 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

Appendix C

List of Definitions

1	Weak Principle of Induction	7	23	Anti-symmetric	27
2	Strong Principle of Induc-		24	Equivalence Relation . . .	28
	tion	8	25	Path	29
3	Least Number Principle .	10	26	Strong Connectedness . .	29
4	Your Ancestors	17	27	Weak Connectedness . . .	30
5	Subset	18	28	Totality	30
6	Superset	18	29	Inverse	30
7	Disjoint	19	30	Complement	31
8	Intersection	19	31	Ordering	31
9	Relative Complement . . .	19	32	Well-ordering	32
10	Union	19	33	Unary Function	33
11	Power Set	21	34	Function	34
12	Ordered Pair	22	35	Domain	34
13	Kuratowski Ordered Pair .	22	36	Range	34
14	Ordered n -tuple	24	37	Partial and Total	34
15	Ordered 1-tuple	25	38	Operator	35
16	Binary Relation	25	39	Commutative	35
17	Cartesian Product	25	40	Associative	35
18	n -place Relation	26	41	Into, Onto, etc.	36
19	Finite Directed Graph . .	26	42	Inverse	37
20	Reflexive	26	43	Finite	38
21	Transitive	27	44	Enumeration	38
22	Symmetric	27	45	[Un]Countable	39

46	Equinumerosity	39	79	nand and nor	93
47	Cardinality	39	80	Duality	94
48	\leq	39	81	Compactness	101
49	String	47	82	Finite Satisfiability	102
50	Sentence Letters	48	83	Recursivity	108
51	Sentences of \mathcal{L}_1	48	84	Recursive Enumerability	109
52	Arity	49	85	Atomic Formula	137
53	Scope	49	86	Formula	137
54	Main Connective	50	87	Free and Bound Variable Occurrences	138
55	Complexity	52	88	Sentence of \mathcal{L}_2	138
56	Valuation	60	89	\mathcal{L}_2 -Structure	139
57	\mathcal{L}_1 -Structure	60	90	Variable Assignment	140
58	\mathcal{L}_1 -valuation	60	91	Satisfaction	140
59	Agreement of Structures	61	92	Truth	142
60	Satisfaction	64	93	Validity in \mathcal{L}_2	143
61	Entailment	64	94	Definite Descriptions	171
62	Tautology	64	95	Direct Reference	192
63	Properties of arguments	66	96	Rigid Designator	193
64	Equivalence	70			
65	Constituent sentence	74			
66	Uniform Substitution	74			
67	Substitution Instance	74			
68	Truth Function	78			
69	Expression	80			
70	Truth-functional Connect- ive	81			
71	Alternative \mathcal{L}_1 -valuation	82			
72	Base	83			
73	Arbitrary Conjunction	85			
74	Literal	85			
75	Disjunctive Normal Form (DNF)	85			
76	Expressive Adequacy	89			
77	Functional Completeness	89			
78	Maximum and Minimum	90			

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 Ślupecki, 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.
- 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/.

BIBLIOGRAPHY

- 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), ‘Scoop-
ing the Loop Snooper: a proof that
the Halting Problem is unsolvable’.
URL [www.ling.ed.ac.uk/~gpullum/
loopsnoop.pdf](http://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 Lo-
gic*. Boston, MA: Harvard Univer-
sity Press.
- Restall, Greg (2000) *An Introduction to
Substructural Logics*. London: Rout-
ledge.
- Richard, Mark (1986) ‘Quotation,
grammar, and opacity’, *Linguistics
and Philosophy* 9: 383–403.
- Russell, Bertrand (1956), ‘On Denot-
ing’. In *Logic and Knowledge*, Lon-
don 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 Stan-
ford Encyclopedia of Philosophy*.
URL [plato.stanford.edu/entries/
logic-classical/](http://plato.stanford.edu/entries/logic-classical/).
- Sider, Theodore (2010), *Logic for Philo-
sophy*. Oxford: Oxford University
Press.
- Stalnaker, Robert C (1975), ‘Indicat-
ive 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, Metamathemat-
ics*, Indianapolis: Hackett, 2nd ed.,
1983, pp. 152–278.
- (1936), ‘On the Concept of
Logical Consequence’. In *Logic*,

BIBLIOGRAPHY

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 \sup '. *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.